



FUNDAÇÃO UNIVERSIDADE DE BRASÍLIA

LÍNGUAS ESTRANGEIRAS APLICADAS AO MULTILINGUISMO E À SOCIEDADE  
DA INFORMAÇÃO

RODRIGO AUGUSTO RACHADEL

**ACESSIBILIDADE *WEB*: UMA ANÁLISE SOBRE SUAS FERRAMENTAS E  
DIRETRIZES**

Brasília  
2017

RODRIGO AUGUSTO RACHADEL

**ACESSIBILIDADE WEB: UMA ANÁLISE SOBRE SUAS FERRAMENTAS E  
DIRETRIZES**

Trabalho de Conclusão de Curso apresentado ao curso de Línguas Estrangeiras Aplicadas ao Multilinguismo e à Sociedade da Informação, da Universidade de Brasília, como requisito para obtenção de grau bacharel.

Orientador: Prof. Dr. Cláudio Corrêa e Castro Gonçalves

Brasília  
2017

*As humans, we collectively avoid a huge percentage of ethical dilemmas by defining them out of existence. We decide that they are beyond our control and are someone else's responsibility. (Peter Morville & Louis Rosenfeld )*

## **AGRADECIMENTOS**

Agradeço primeiramente aos meus pais por serem os principais responsáveis pela minha educação e formação do homem quem sou hoje. Sempre me apoiaram e me incentivaram aos estudos e não mediram esforços para que eu tivesse as melhores oportunidades que poderiam me fornecer. Sempre me lembraram que não importa o quanto se gasta se for em prol dos estudos.

Ao meu companheiro, que esteve com seu abraço e consolo presentes em todas as noites de estresse e ansiedade. Foi agente primordial para minha estabilidade emocional para lidar com este trabalho, bem como sempre me apoiou a continuar e buscar meu melhor como pessoa e como profissional.

Ao meu orientador professor e doutor Cláudio Corrêa que se dispôs a tornar esse trabalho possível. Sem sua experiência, e principalmente paciência, este trabalho seria apenas uma ideia. Toda sua orientação reflete na qualidade deste trabalho.

A minha querida amiga Vitória Doretto, por além de sua energia positiva e companheirismo durante toda minha jornada acadêmica, contribui fortemente para a revisão deste projeto. Pude confiar em seu trabalho de olhos fechados.

A todos meus queridos amigos de Marília/SP e de Natal/RN que mesmo de longe, me deram forças e torceram muito para minha felicidade e para que almejassem meus objetivos. Mesmo com a distância, procuraram se fazer presentes. Em especial, agradeço Abbi Sam por ser principal responsável a me ensinar a ter coragem e orgulho de ser quem sou.

A todos meus professores e colegas que contribuíram e contribuem para a melhoria do nosso curso. Obrigado por todos ensinamentos e apoio. Levarei comigo para toda a vida.

## RESUMO

Mesmo com o avanço tecnológico em prol da difusão de informação, ainda existem muitos *sites* que possuem pouco ou nenhum tipo de acessibilidade *web*. Para este projeto decidimos fazer um estudo de ferramentas para acessibilidade *web*, bem como analisar as diretrizes da arquitetura da informação e do *web design* para uma melhor experiência do usuário, uma vez que o gerenciamento de conteúdo de *site* é muito importante para uma navegabilidade simples e confortável. Ao longo do trabalho, iremos explicar que a acessibilidade na *web* precisa ser constituída tanto pelo desenvolvimento de ferramentas, como aumento e diminuição de fonte, quanto pela estruturação de cada elemento do *site*, seja uma barra de navegação ou um botão. A fim de suprir a carência de *sites* acessíveis, este trabalho tem como objetivo oferecer mais de uma forma para acessibilizar um *site* com a finalidade de favorecer qualquer categoria de desenvolvedor *web*, seja de nível básico, intermediário ou avançado em programação. As ferramentas usadas para cada nível serão respectivamente: um Sistema de Gerenciamento de Conteúdo, um *framework* e uma biblioteca.

**Palavras-chave:** acessibilidade na *web*, arquitetura da informação, experiência do usuário.

## **ABSTRACT**

*Even with a technological breakthrough for the dissemination of information, there are still many websites that have little or no web accessibility. For this project, the author decided to do a study and development of tools for web accessibility as well as analyze the guidelines of information architecture and web design for a better user experience, since site content management is very important for a simple and comfortable navigation. The author will explain that web accessibility needs to be constituted by the development of tools, such as increase and decrease of font, as well as by the structuring of each element of the site, such a navigation bar or a button. In order to fill the lack of accessible websites, this work aims to offer more than one way to make a website accessible in order to favor any category of web developer, whether basic, intermediate or advanced programming. The tools used for each level will be respectively a Content Management System, a framework and a library.*

**Key-words:** *web accessibility, information architecture, user experience.*

## LISTA DE FIGURAS

FIGURA 1 - Número de usuários na internet .....	14
FIGURA 2 – Cabeçalho do site do World Wide Web Consortium .....	17
FIGURA 3 – Página inicial do site do World Wide Web Consortium em português.....	17
FIGURA 4 – Página ‘Arquitetura Web’ do site do World Wide Web Consortium em inglês .....	18
FIGURA 5 – Exemplo de Top Navigation .....	23
FIGURA 6 – Exemplo de Local Navigation.....	24
FIGURA 7 – Exemplo de Contextual Navigation .....	25
FIGURA 8 – Top dez línguas na internet em 2013 .....	30
FIGURA 9 – Cabeçalho da página do site do Acessibilidade Legal .....	31
FIGURA 10 – Cabeçalho da página do site do Acesso Para Todos .....	31
FIGURA 11 – Cabeçalho da página do site da APABB .....	32
FIGURA 12 – Cabeçalho da página do site da Vida Mais Livre .....	32
FIGURA 13 – Cabeçalho da página site do Governo do Brasil.....	32
FIGURA 14 – Fluxograma do site .....	37
FIGURA 15 – Identificação da Top Navigation no fluxograma do site.....	39
FIGURA 16 – Esboço do site.....	40
FIGURA 17 – Página do plug-in Google Language Translator.....	45
FIGURA 18 – Página de widgets do WordPress .....	46
FIGURA 19 – Página inicial do site exemplo em WordPress .....	47
FIGURA 20 – Página do plug-in Hand Talk - WordPress.....	48
FIGURA 21 – Caixa de propriedades do plug-in Hand Talk - WordPress .....	49
FIGURA 22 – Intérprete virtual do plug-in Hand Talk - WordPress.....	50
FIGURA 23 – Página do plug-in WP Accessibility Helper .....	51
FIGURA 24 – Página de configuração do plug-in WP Accessibility Helper.....	52
FIGURA 25 – Resultado final do site construído em WordPress .....	53
FIGURA 26 – Código CDN do Bootstrap .....	56
FIGURA 27 – Arquivos de instalação do plug-in Paypal Accessibility.....	57
FIGURA 28 – Arquivos do plug-in Paypal Accessibility na pasta JavaScript .....	57
FIGURA 29 – Código CDN do plug-in Paypal Accessibility .....	58
FIGURA 30 – Código do botão do Bootstrap.....	59
FIGURA 31 – Interface gráfico do botão do Bootstrap.....	59
FIGURA 32 – Código do modal do Bootstrap.....	60
FIGURA 33 – Interface gráfica do modal do Bootstrap.....	61
FIGURA 34 – Código do tooltip do Bootstrap .....	62
FIGURA 35 – Interface gráfica do tooltip do Bootstrap.....	62
FIGURA 36 – Interface gráfica do dropdown do Bootstrap .....	63
FIGURA 37 – Código do dropdown do Bootstrap .....	64
FIGURA 38 – Interface gráfica do tab do Bootstrap .....	65
FIGURA 39 – Código do tab do Bootstrap .....	66
FIGURA 40 – Código do conteúdo do tab do Bootstrap .....	66
FIGURA 41 – Interface gráfica do collapse do Bootstrap .....	67
FIGURA 42 – Código do collapse do Bootstrap sem o plug-in Paypal Accessibility .....	68
FIGURA 43 – Código do collapse do Bootstrap com o plug-in Paypal Accessibility.....	69
FIGURA 44 – Código do conteúdo do collapse do Bootstrap.....	69
FIGURA 45 – Barra de navegação do site em Bootstrap .....	70
FIGURA 46 – Arquivos das páginas em inglês, francês e espanhol.....	71

FIGURA 47 – Código do atributo LANG para o multilinguismo com Bootstrap .....	71
FIGURA 48 – Código do dropdown do Bootstrap .....	72
FIGURA 49 – Código de pular para o conteúdo principal do Bootstrap.....	73
FIGURA 50 – Código CDN do jQuery .....	75
FIGURA 51 – Pasta com extensão .js para código jQuery .....	76
FIGURA 52 – Código de importação do arquivo app.js.....	76
FIGURA 53 – Interface gráfico do dropdown do Bootstrap para jQuery.....	76
FIGURA 54 – Código jQuery para botões de sublinhar e destacar links e contraste de cor de fundo .	77
FIGURA 55 – Código jQuery para botão de aumento de fonte .....	78
FIGURA 56 – Código jQuery para botão diminuição de fonte .....	78
FIGURA 57 – Caixa de propriedades do token de Hand Talk .....	79



## **LISTA DE QUADROS**

QUADRO 1 – Resultado da capacidade de acessibilidade do WordPress.....	53
QUADRO 2 – Resultado da capacidade de acessibilidade do Bootstrap .....	73
QUADRO 3 – Resultado da capacidade de acessibilidade do jQuery .....	79
QUADRO 4 – Resultado final da capacidade de acessibilidade do WordPress, Bootstrap e jQuery...	80

## LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
AI	Arquitetura da Informação
APABB	Associação Brasileira de Normas Técnicas Associação de Pais, Amigos e Pessoas com Deficiência, de Funcionários do Banco do Brasil e da Comunidade
CDN	<i>Content Delivery Network</i>
CMS	<i>Content Management System</i>
CSS	<i>Cascading Style Sheet</i>
DOM	<i>Document Object Model</i>
HTML	<i>Hypertext Markup Language</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
JS	JavaScript
LBI	Lei Brasileira de Inclusão
LIBRAS	Língua Brasileira de Sinais
NVDA	<i>NonVisual Desktop Access</i>
PHP	<i>PHP: Hypertext Preprocessor</i>
SASS	<i>Syntactically Awesome Stylesheets</i>
TDI	<i>Tabbed Document Interface</i>
WAI	<i>Web Accessibility Initiative</i>
WP	WordPress
WWW	<i>World Wide Web</i>
W3C	<i>World Wide Web Consortium</i>

## SUMÁRIO

1. INTRODUÇÃO	13
2. JUSTIFICATIVA	15
3. OBJETIVO	18
4. REFERENCIAL TEÓRICO	19
4.1. SOCIEDADE DA INFORMAÇÃO	19
4.2. ARQUITETURA DA INFORMAÇÃO E EXPERIÊNCIA DO USUÁRIO	20
4.2.1. IMPORTÂNCIA E CONCEITOS DA ARQUITETURA DA INFORMAÇÃO	22
4.2.2. ACESSIBILIDADE WEB	26
4.2.2.1. MULTILINGUISMO	29
5. SITES CORRELATOS	30
6. BOAS PRÁTICAS DE ACESSIBILIDADE NA WEB	33
6.1. LEVANTAMENTO DE REQUISITOS E FLUXOGRAMA	36
6.1.1. CONSTRUINDO UM FLUXOGRAMA	37
6.1.2. A PROGRAMAÇÃO E ALGUNS CONCEITOS	40
7. SISTEMA DE GERENCIAMENTO DE CONTEÚDO (CMS)	41
7.1. WORDPRESS	42
7.1.1. INSTALANDO OS PLUG-INS	43
7.1.1.1. GOOGLE LANGUAGE TRANSLATOR	44
7.1.1.2. HAND TALK - WORDPRESS	47
7.1.1.3. WP ACCESSIBILITY HELPER	50
7.1.2. CONCLUSÕES DO CMS	53
8. FRAMEWORK	54
8.1. BOOTSTRAP	55
8.1.1. INSTALANDO O BOOTSTRAP	55
8.1.2. PLUG-IN PAYPAL ACCESSIBILITY	57
8.1.2.1. MODAL E TOOLTIP	58
8.1.2.2. DROPDOWN	63
8.1.2.3. TAB	65
8.1.2.4. COLLAPSE	67
8.1.3. MULTILINGUISMO COM BOOTSTRAP	69
8.1.4. PULAR PARA O CONTEÚDO	72

8.1.5. CONCLUSÕES DO FRAMEWORK	73
9. BIBLIOTECAS	74
9.1. JQUERY	74
9.1.1. INSTALANDO JQUERY	75
9.1.2. SUBLINHAR E DESTACAR LINKS	77
9.1.3. BOTÃO PARA AUMENTAR E DIMINUIR FONTE	77
9.1.4. JANELA DE LIBRAS	78
9.1.5. CONCLUSÕES DA BIBLIOTECA	79
10. CONSIDERAÇÕES FINAIS	80
REFERÊNCIAS BIBLIOGRÁFICAS	82

## 1. INTRODUÇÃO

Sabemos que a invenção do computador e da *internet* foi um marco para o surgimento do que chamamos hoje de Sociedade da Informação. A ferramenta alavancou a sociedade para um nível de difusão de informação que há alguns anos não poderíamos imaginar.

De forma simples, o computador é “um sistema programável, que permite armazenar, recuperar e processar dados (NUNES, s.d., p. 4)”. Neste processo de armazenamento e processamento de dados, a máquina possui um grande e longo procedimento para nos trazer a informação que requisitamos. Desde resultados de cálculos, que foi sua finalidade *a priori* com as invenções do Ábaco (2500 ac), Calculadora de Pascal (1642) e Tabuladora Holerith (1890), até o compartilhamento de entretenimento, cultura e mídias digitais (NUNES, s.d.).

O acesso a esse conteúdo não seria possível se não houvesse a invenção da *internet*. É por meio dela que temos um vasto conhecimento digital a nosso dispor. Carlos Alberto Rosa Ferreira, professor auxiliar da Faculdade de Motricidade Humana da Universidade de Lisboa, define *internet* em seu artigo *Internet* como:

capacidades de transmissão a nível mundial, um mecanismo de disseminação de informação, e um meio para colaboração e interação entre indivíduos e os seus computadores sem considerar a sua posição geográfica. (2002, p. 1)

Essa rede de transmissão é tão poderosa, que segundo o Acervo Digital da Universidade Federal Rural de Pernambuco<sup>1</sup>, a *internet* é o meio de comunicação que se expandiu no mundo mais rápido que a televisão, o rádio e o telefone, por exemplo. Segundo o acervo, para atingir 50 milhões de usuários, o rádio levou 38 anos, a televisão levou 13 anos, o telefone levou 16 anos e a *internet* levou apenas quatro anos.

Como resultado desta expansão, os dados da *Internet World Stats*<sup>2</sup>, disponíveis na Figura 1, informam que em junho de 2017, mais do que a população mundial já possui acesso à *internet*.

---

<sup>1</sup> Disponível em: <<http://www.ead.ufrpe.br/acervo-digital-eadtec/node/665>>. Acesso em: 18 nov. 2017.

<sup>2</sup> Disponível em: <<http://www.internetworldstats.com/stats.htm>>. Acesso em: 18 nov. 2017.

**FIGURA 1** - Número de usuários na internet

WORLD INTERNET USAGE AND POPULATION STATISTICS JUNE 30, 2017 - Update						
World Regions	Population ( 2017 Est.)	Population % of World	Internet Users 30 June 2017	Penetration Rate (% Pop.)	Growth 2000-2017	Internet Users %
<a href="#">Africa</a>	1,246,504,865	16.6 %	388,376,491	31.2 %	8,503.1%	10.0 %
<a href="#">Asia</a>	4,148,177,672	55.2 %	1,938,075,631	46.7 %	1,595.5%	49.7 %
<a href="#">Europe</a>	822,710,362	10.9 %	659,634,487	80.2 %	527.6%	17.0 %
<a href="#">Latin America / Caribbean</a>	647,604,645	8.6 %	404,269,163	62.4 %	2,137.4%	10.4 %
<a href="#">Middle East</a>	250,327,574	3.3 %	146,972,123	58.7 %	4,374.3%	3.8 %
<a href="#">North America</a>	363,224,006	4.8 %	320,059,368	88.1 %	196.1%	8.2 %
<a href="#">Oceania / Australia</a>	40,479,846	0.5 %	28,180,356	69.6 %	269.8%	0.7 %
<b>WORLD TOTAL</b>	<b>7,519,028,970</b>	<b>100.0 %</b>	<b>3,885,567,619</b>	<b>51.7 %</b>	<b>976.4%</b>	<b>100.0 %</b>
NOTES: (1) Internet Usage and World Population Statistics updated as of June 30, 2017. (2) CLICK on each world region name for detailed regional usage information. (3) Demographic (Population) numbers are based on data from the <a href="#">United Nations Population Division</a> . (4) Internet usage information comes from data published by <a href="#">Nielsen Online</a> , by ITU, the <a href="#">International Telecommunications Union</a> , by GfK, by local ICT Regulators and other reliable sources. (5) For definitions, navigation help and disclaimers, please refer to the <a href="#">Website Surfing Guide</a> . (6) Information from this site may be cited, giving the due credit and placing a link back to <a href="http://www.internetworldstats.com">www.internetworldstats.com</a> . Copyright © 2017, Miniwatts Marketing Group. All rights reserved worldwide.						

Fonte: Internet World Stats.

À luz destes dados, podemos perceber a importância de trabalharmos a favor da difusão de informação. Muitos destes usuários são pessoas com alguma necessidade especial, motora ou cognitiva ou, então, não conhecedoras do idioma em que o conteúdo está escrito e que, infelizmente, encontram dificuldades para acessar essa gama de informação que está disponível a todos. Em grandes redes sociais, páginas governamentais e comerciais estão carentes de ferramentas que viabilizam de forma eficaz a usabilidade e navegabilidade de usuários para com seu conteúdo. O que pode agravar esse cenário é a falta de informação de como se dá o processo de desenvolvimento de ferramentas de acessibilidade e o desconhecimento das boas práticas e diretrizes sobre o gerenciamento de conteúdo *web* a favor da experiência do usuário.

Este trabalho iniciará com sua justificativa, que é uma análise da importância da criação e aplicação de ferramentas que proporcionam conforto e comodidade ao acesso de informação para qualquer pessoa, seja ouvinte, surda, ensurdecida, vidente, cega ou estrangeira. No capítulo três, serão expostos os objetivos do projeto, como o esclarecimento das indicações e instruções do gerenciamento do conteúdo, com foco no desenvolvimento das ferramentas que suprem a carência de acessibilidade na *web*. No capítulo quatro será apresentado o referencial teórico que se dará com citações e referências sobre sociedade da informação, arquitetura da informação e acessibilidade na *web* para melhor reflexão do trabalho, bem como a consolidação da lista de *features* de acessibilidade que serão

desenvolvidas. Neste mesmo capítulo, teremos um subitem sobre a importância do multilinguismo como ferramenta de acessibilidade *web*. No capítulo sete será elaborada a metodologia que é a mostra do desenvolvimento das *features* organizada em nível de complexidade para usuários de básico, intermediário e avançado conhecimento em programação por meio de Sistema de Gerenciamento de Conteúdo, *framework* e biblioteca respectivamente. No capítulo 10, a conclusão, haverá uma análise sobre o desempenho das três formas de criação de *features*, bem como suas facilidades e dificuldades encontradas entre elas.

## 2. JUSTIFICATIVA

A acessibilidade é importante no trato da sociedade da informação. A difusão da informação hoje possui um rico valor no desenvolvimento do indivíduo. Felizmente, existem algumas leis que viabilizam isso. Naturalmente, todas as pessoas possuem o direito de usufruir de qualquer informação pública contida na web, assim como diz a Lei Brasileira de Inclusão:

Art. 63. É obrigatória a acessibilidade nos sítios da internet mantidos por empresas com sede ou representação comercial no País ou por órgãos de governo, para uso da pessoa com deficiência, garantindo-lhe acesso às informações disponíveis, conforme as melhores práticas e diretrizes de acessibilidade adotadas internacionalmente (LBI, Lei 13.146/15. Art. 63).

Entretanto, mesmo com algumas políticas públicas a favor da acessibilidade, ainda existem muitos *sites* sem nenhum tipo de trabalho que facilite o acesso de algumas pessoas. Segundo o censo do Instituto Brasileiro de Geografia e Estatística (IBGE) de 2010, 45,6 milhões de pessoas declararam ter ao menos um tipo de deficiência, o que corresponde a 23,9% da população brasileira. É um número muito grande de pessoas que possuem dificuldade para alcançar o direito de ter acesso a informação contida na *internet*.

Quando tornamos um *site* multilíngue para estrangeiros, ou audiodescrevemos as imagens para um cego, por exemplo, estamos expandindo o número de usuários que acessam o conteúdo de nosso *site*. Em outras palavras, quando trabalhamos com a acessibilidade em nossa página *web*, que pode ser um *e-commerce*, por exemplo, estamos investindo no número de acessos e aumento de poder de compra dos nossos produtos. Imagine quantos clientes em potencial poderia estar perdendo só porque não possui um *site* acessível.

No entanto, há um equívoco dos desenvolvedores ao achar que a acessibilidade é cara e por isso deixar o serviço para depois. É importante entender que aplicar as ferramentas de acessibilidade em um *site* depois de sua criação é bem mais caro do que desenvolvê-lo desde o início com as ferramentas. Pensem na construção de um prédio: não faz sentido quebrar todas as escadas para mais tarde colocar as rampas, certo? É inteligente e benéfico desenvolver um *site* desde o início com as ferramentas adequadas bem como as diretrizes em prol da acessibilidade e torna-se necessário pela importância de fazer um investimento a longo prazo para evitar futuros gastos extras.

Por esses motivos, um estudo sobre como desenvolver as ferramentas de acessibilidade para *web* torna-se indispensável. Ao longo deste trabalho, iremos abordar também a importância da Arquitetura da Informação e da Experiência do Usuário na navegabilidade dos *sites*. Iremos perceber que um *site* que contém unicamente as ferramentas de acessibilidade não é totalmente efetivo na questão de difusão de informação. A organização dos conteúdos da página merece sua devida atenção.

O grande número de informação na *internet* promove ao usuário muitas possibilidades como interação, comunicação, agilidade e praticidade (PASSOS, 2017) e para lidar com essas características, é necessário uma organização e padronização para o tratamento desta informação. Para obter um conhecimento preciso sobre estes padrões para *web* foi feita uma pesquisa no *site* do *World Wide Web Consortium* (W3C). O W3C é a principal organização internacional de padrões para a *World Wide Web* (abreviada WWW ou W3). Atualmente, Tim Berners-Lee, além de fundador, lidera o consórcio que é constituído por organizações que visam trabalhar em conjunto na criação de padrões para a *web*. O W3C também tem como foco a educação e a divulgação, desenvolve *software* e serve como um fórum aberto para discussão sobre a *web*.

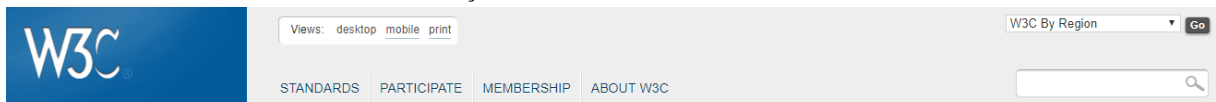
Durante a pesquisa sobre padrões de arquitetura *web*, não levou muito tempo para que encontrássemos dificuldades com a navegabilidade do *site*, uma vez que o autor deste trabalho não possui a língua inglesa como primeira língua. Mesmo em um *site* tão importante que visa experiência do usuário como o W3C<sup>3</sup>, ainda são encontrados problemas de acessibilidade multilíngue e são situações como essas que este trabalho tem como propósito levantar questionamentos da necessidade da criação de ensejos de acessibilidade. Veremos nas imagens abaixo um exemplo dos problemas encontrado na usabilidade do *site*.

---

<sup>3</sup> Disponível em: <<https://www.w3.org/>>. Acesso em: 25 set. 2017.



**FIGURA 2** – Cabeçalho do site do World Wide Web Consortium



Fonte: Elaborado pelo autor (captura de tela).

No momento que o usuário seleciona o idioma de sua escolha no canto superior direito, a ferramenta funciona corretamente. O *site* é naturalmente em inglês e depois da seleção do usuário, o seu conteúdo da página inicial apareceu em português.

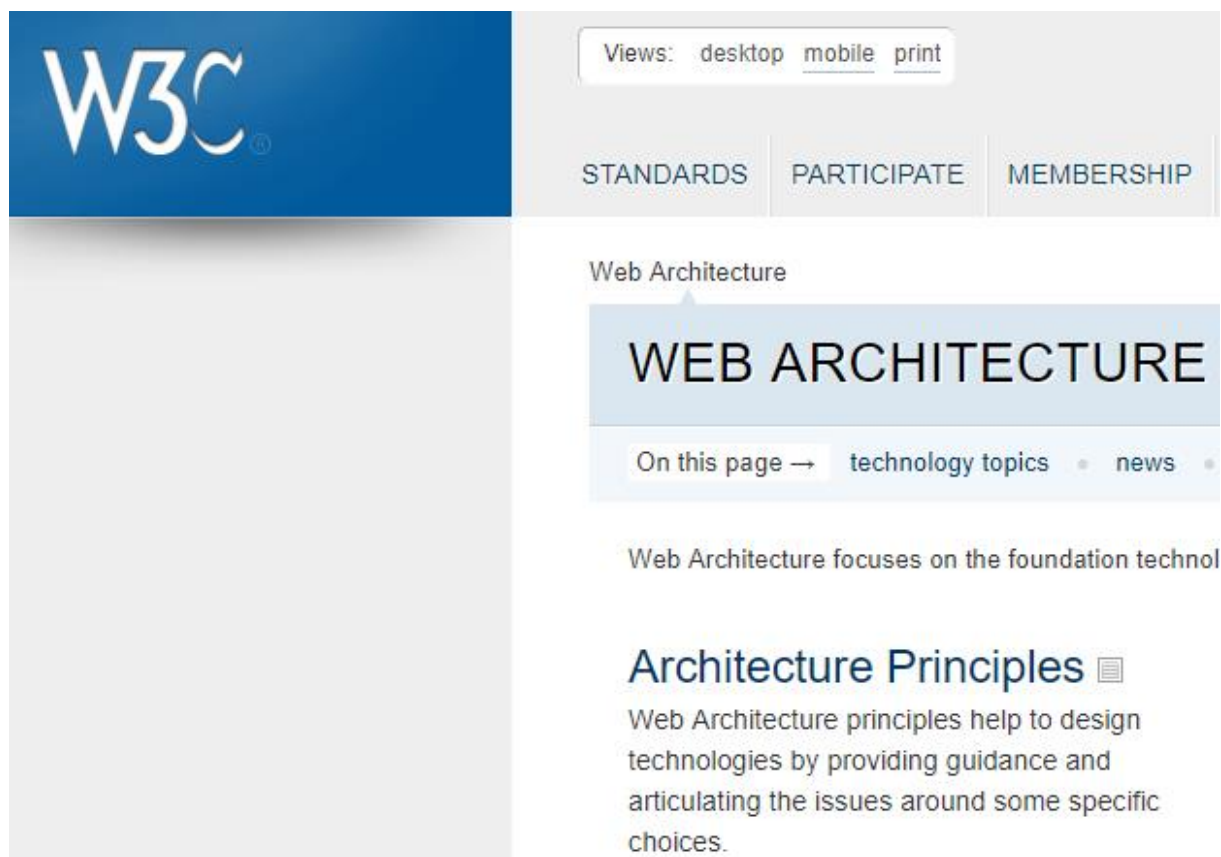
**FIGURA 3** – Página inicial do site do World Wide Web Consortium em português



Fonte: Elaborado pelo autor (captura de tela).

O problema começa no momento que o usuário inicia a navegação pela barra lateral. Ao selecionar o item “Arquitetura Web”, por exemplo, a página a ser exibida é em inglês e não em português como o usuário havia escolhido. Veja na imagem abaixo como o próprio *site* de uma das maiores organizações de padronizações da *web* também precisa ter seu conteúdo revisado para atender as demandas de todos os usuários.

**FIGURA 4** – Página ‘Arquitetura Web’ do site do World Wide Web Consortium em inglês



Fonte: Elaborada pelo autor (captura de tela).

A acessibilidade na *web* é uma área que vem tomando espaço gradualmente nas mídias e principalmente na academia e é crucial que continuemos com esses estudos. São pelos motivos mencionados neste capítulo que a instrução de quais são as boas práticas e diretrizes para uma boa arquitetura da informação e de como desenvolver as ferramentas de acessibilidade se fazem importantes. Com tanta informação a sua posse, a *internet* é um meio de comunicação que *exige* acessibilidade para adequar-se a todos. É muito conhecimento para ficar ao alcance de poucos.

### 3. OBJETIVO

O objetivo deste trabalho é fazer um estudo sobre o desenvolvimento de ferramentas em prol da acessibilidade na *web*, bem como abordar as diretrizes de navegabilidade e experiência do usuário por meio da Arquitetura da Informação e do *Web Design*. Como um produto de análise feita em *sites* correlatos, criaremos uma lista de *features* que um *site* precisa ter para ser considerado acessível. Será utilizada como métrica de complexidade para usuários de baixo, intermediário e alto conhecimento em programação

ferramentas como Sistema de Gerenciamento de Conteúdo, *framework* e bibliotecas respectivamente.

## 4. REFERENCIAL TEÓRICO

### 4.1. SOCIEDADE DA INFORMAÇÃO

Segundo Luis Borges Manuel Gouveia, professor catedrático da Universidade Fernando Pessoa, em Portugal, explica em seu artigo *Sociedade da Informação* (2004), uma sociedade da informação é uma sociedade onde a criação, uso, integração, distribuição e manipulação de informações é uma atividade econômica, política e cultural significativa. Seus principais impulsionadores são as tecnologias digitais de informação e comunicação, que resultaram em uma explosão de informação e estão mudando profundamente todos os aspectos da organização social, incluindo economia, educação, saúde, governo e democracia.

Dentro deste contexto, nada mais justo que todas as pessoas possam estar inseridas e acompanhar essas transformações que é visto como sucessoras da Revolução Industrial. Conforme Sérgio Sobral Nunes, mestre em Gestão da Informação na Universidade do Porto, afirma em seu artigo *A Acessibilidade na Internet no Contexto da Sociedade da Informação* (2002, p. 4), uma pessoa ter uma deficiência representa antes de mais nada ser uma “pessoa”, um indivíduo, “e a sociedade reconhece a responsabilidade social de encontrar respostas adequadas às suas necessidades especiais [...]”. O gradual aumento das tecnologias na nossa sociedade nos mostra que uma vez que caminhamos para uma sociedade da informação, temos capacidade de trabalhar em prol da inclusão das pessoas nesta sociedade.

De acordo com Nunes (2002), é possível que com uma transição dos serviços governamentais para a *web*, existirá um grande risco de exclusão de uma grande parcela da população se não forem tomadas medidas cabíveis em prol da acessibilidade da informação, mas se essas ferramentas de acessibilidade forem bem trabalhadas, há grandes chances de diminuirmos esse abismo que existe na sociedade em relação aos cidadãos com necessidades especiais.

A acessibilidade na *web* causa um impacto muito positivo para a sociedade da informação e principalmente na vida das pessoas. Como Jorge Fernandes e Francisco Godinho (2003, p. 13) afirmaram em *Acessibilidade aos sítios Web da AP para cidadãos com necessidades especiais*: “Para a maioria das pessoas, a tecnologia torna a vida mais fácil. Para uma pessoa com necessidades especiais, a tecnologia torna as coisas possíveis”. Neste cenário, como agente de difusão de conhecimento, este trabalho visa contribuir um pouco para

essa sociedade que vem acumulando tanta informação gradualmente e que precisa ser tratada e manuseada adequadamente para estar ao alcance de todos.

## 4.2. ARQUITETURA DA INFORMAÇÃO E EXPERIÊNCIA DO USUÁRIO

Iniciaremos esse subitem com a definição de Arquitetura da Informação (AI). Peter Morville e Louis Rosenfeld em *Information Architecture for the World Wide Web* a definem como:

1. projeto estrutural de um ambiente de informação compartilhado; 2. a combinação de organização, rotulagem, busca e sistemas de navegação dentro de *sites* e *intranets*; 3. a arte e a ciência de moldar produtos e experiências de informação para suportar a usabilidade e a *findability*; 4. uma disciplina emergente e uma comunidade de prática focada em trazer princípios de *design* e arquitetura para a paisagem digital (2006, p. 4, tradução nossa<sup>4</sup>).

É um grande equívoco esperar por uma única definição, algo que dominasse toda a essência da área. Morville e Rosenfeld (2006) procuraram explicar com mais detalhes a AI em quatro características: informação; estruturação, organização e rotulagem; busca e gerenciamento; e arte e ciência.

Eles explicam que a fim de diferenciar a arquitetura da informação e gerenciamento de dados de conhecimento, utiliza-se o termo *informação*. Por um lado, os dados são considerados fatos e números, assim quando se relacionam com um banco de dados, eles geram respostas específicas para questões específicas. Por outro lado, o conhecimento é o material que se encontra na mente das pessoas, o que é incentivado pelos gerentes de conhecimento a criar ferramentas e processos e para encorajar as pessoas a compartilhar este conhecimento. A *informação* se encontra no meio dos dois. Com sistemas de informação, para uma determinada questão não há uma única resposta, na maioria das vezes. Arquitetos da Informação preocupam-se com *sites*, documentos, aplicativos de software e imagens, bem como metadados como termos usados para descrever e representar objetos de conteúdo, como documentos, pessoas, processos e organizações.

Os autores afirmam que a *estruturação* compreende a definição dos níveis adequados de granularidade para o conjunto de informações no *site* e decide como relacioná-las umas com as outras. *Organizar* envolve o agrupamento desses conjuntos em categorias

---

<sup>4</sup>1. The structural design of shared information environments. 2. The combination of organization, labeling, search, and navigation systems within web sites and intranets. 3. The art and science of shaping information products and experiences to support usability and findability. 4. An emerging discipline and community of practice focused on bringing principles of design and architecture to the digital landscape.

relevantes e específicas. *Rotulagem* significa descobrir como chamar essas categorias e a série de *links* de navegação que levam a elas.

Eles seguem explicando que a *capacidade de busca* é um fator crítico para a usabilidade em geral. O *site* será ineficiente se os usuários não conseguem encontrar o que precisam por meio de uma combinação de busca e navegação. Contudo, centrar o *design* unicamente no usuário não é suficiente. Quem gerencia as informações, como organizações e pessoas, também é importante. As necessidades dos usuários e os objetivos do mercado devem ser equilibrados pela Arquitetura da Informação.

Na quarta e última característica, os autores afirmam que a rigidez do *método científico* para a análise das necessidades dos usuários e comportamentos de busca de informações estão sendo fortemente trazidas pelas disciplinas como engenharia de usabilidade e etnografia. Gradualmente estamos preparados para estudar os padrões de uso e subsequentemente prover melhorias para *sites*, contudo a prática da arquitetura da informação nunca será reduzida aos números. Existe muita ambiguidade e complexidade. Além de estarem dispostos a correr riscos e confiar na própria intuição, os arquitetos da informação devem contar com experiência e criatividade. Esta é a "arte" da arquitetura da informação.

Se fizéssemos uma analogia ao mundo físico, um *site* poderia muito bem ser comparado a uma livraria (MORVILLE; ROSENFELD, 2006). Se não houver organização, caso haja livros jogados nas mesas e cadeiras e sem etiquetas, aquele ambiente será um imenso desafio aos seus clientes para conseguir achar o que buscam. Isso se aplica em *site*, se não houver organização e estruturação e inclusive etiquetas – no código semântico –, não haverá acessibilidade para o usuário conseguir com autonomia alcançar aquilo que procura, mas claro, existem grandes diferenças entre livrarias e *sites*.

Algumas pessoas podem se equivocar ao pensar que arquitetura da informação é o mesmo que *design* gráfico. Não, elas não são. Morville e Rosenfeld (2006) afirmam que tradicionalmente, um *designer* gráfico é responsável por todos os aspectos da comunicação visual, desde o *design* de logotipos corporativos e identidades até o *layout* de páginas individuais.

Os arquitetos da informação se concentram mais no conceito da acessibilidade. Eles possuem um papel de gerenciadores de conteúdo. São os responsáveis por organizar de forma mais ampla os componentes do *site*, o posicionamento, a visibilidade e a estrutura. Apesar de tanto os arquitetos da informação quanto os *designers* gráficos darem a devida atenção à composição das cores em prol da acessibilidade da informação, o papel do primeiro é muito além de identidades de *layout* ou construção de logo.

#### 4.2.1. IMPORTÂNCIA E CONCEITOS DA ARQUITETURA DA INFORMAÇÃO

A arquitetura da informação é uma área que nos mostra a importância da construção e estruturação do componente de qualquer *site*. Por meio dela entendemos que não basta uma página *web* conter suas ferramentas de acessibilidade se o seu conteúdo não segue uma lógica hierárquica. Um *site* sem uma boa arquitetura da informação, bem como um ótimo *design*, de nada valerá. Neste subitem, podemos perceber que a AI é uma área rica e de grande valor para a experiência do usuário e que há algumas especificidades dentro de seu conteúdo.

Para medir a importância da Arquitetura da Informação, Morville e Rosenfeld (2006) levantaram alguns dados e decidiram fazer as seguintes perguntas para os leitores de seu livro *Information Architecture for the World Wide Web*:

[...] Quanto custa se cada funcionário da sua empresa gastar mais cinco minutos por dia se esforçando para encontrar respostas na sua *intranet*? Quanto custa frustrar o seu cliente com uma pobre organização do *site*? [...] Quantos clientes você perde por eles não encontrarem em seu *site* o produto que eles procuram e acabam procurando em outros *sites*? [...] quanto custa projetar e construir um *site*? Quanto custa reconstruí-lo seis meses depois porque não suporta *findability* e acessibilidade? (2006, p. 11, tradução nossa<sup>5</sup>).

Para a primeira pergunta, os autores do livro afirmaram que Jakob Nielsen (um consultor de usabilidade na web com Ph.D. na interação homem-homem da Universidade Técnica da Dinamarca em Copenhague) divulgou o fato de que os custos do *design* de sistema de navegação deficiente em uma grande empresa podem somar um prejuízo de milhões de dólares de produtividade perdida dos funcionários.

Os autores dão início aos seus estudos definindo um dos componentes mais importantes da arquitetura de um *site*: a barra de navegação. De forma simples, eles definem a barra de navegação como:

[...] uma coleção distinta de links de hipertexto que conectam uma série de páginas, permitindo o movimento entre elas. Eles podem suportar a navegação global, local e contextual. Você pode implementar a navegação

---

<sup>5</sup>[...] What does it cost if every employee in your company spends an extra five minutes per day struggling to find answers on your intranet? What is the cost of frustrating your customers with a poorly organized web site? [...] How many customers do you lose because they can't find the product they want on your web site? [...] What does it cost to design and build a web site? How much does it cost to redo it six months later because it doesn't support findability or doesn't scale?

de várias maneiras, usando texto ou gráficos, *pull-downs*, *pop-ups*, *rollovers*, menus em cascata, e assim por diante (2006, p. 129, tradução nossa<sup>6</sup>).

Na arquitetura da informação, como já vimos, é muito importante levarmos em consideração o posicionamento dos elementos que compõem a estrutura do *site*. Morville e Rosenfeld afirmam que devemos nos atentar a certos questionamentos no momento em que começamos a trabalhar com navegação global, local e contextual:

A barra de navegação local funciona melhor no topo da página ou é melhor pelo lado esquerdo? Devemos usar *pull-downs*, *pop-ups* ou menus em cascata para reduzir o número de cliques necessários? Não é melhor usar a convenção de cores do *link* azul / vermelho? (2006, p. 117, tradução nossa<sup>7</sup>).

Todas essas questões são importantes no momento da construção da acessibilidade. Ao projetar um sistema de navegação, é importante considerar o ambiente em que o sistema irá existir.

Segundo os autores, *Navegação Global* é uma estrutura de navegação que se faz existente em todas as páginas de uma página *web*.

Muitas vezes, é implementada sob a forma de uma barra de navegação na parte superior de cada página. Estes sistemas de navegação em todo o *site* permitem o acesso direto a áreas e funções chave, independentemente de o usuário viajar na hierarquia do site (2006, p. 122, tradução nossa<sup>8</sup>).

Veja a Figura 5, retirada do livro de Morville e Rosenfeld (2006).

**FIGURA 5** – Exemplo de Top Navigation



Fonte: *Information Architecture for the World Wide Web* (2006, p. 122).

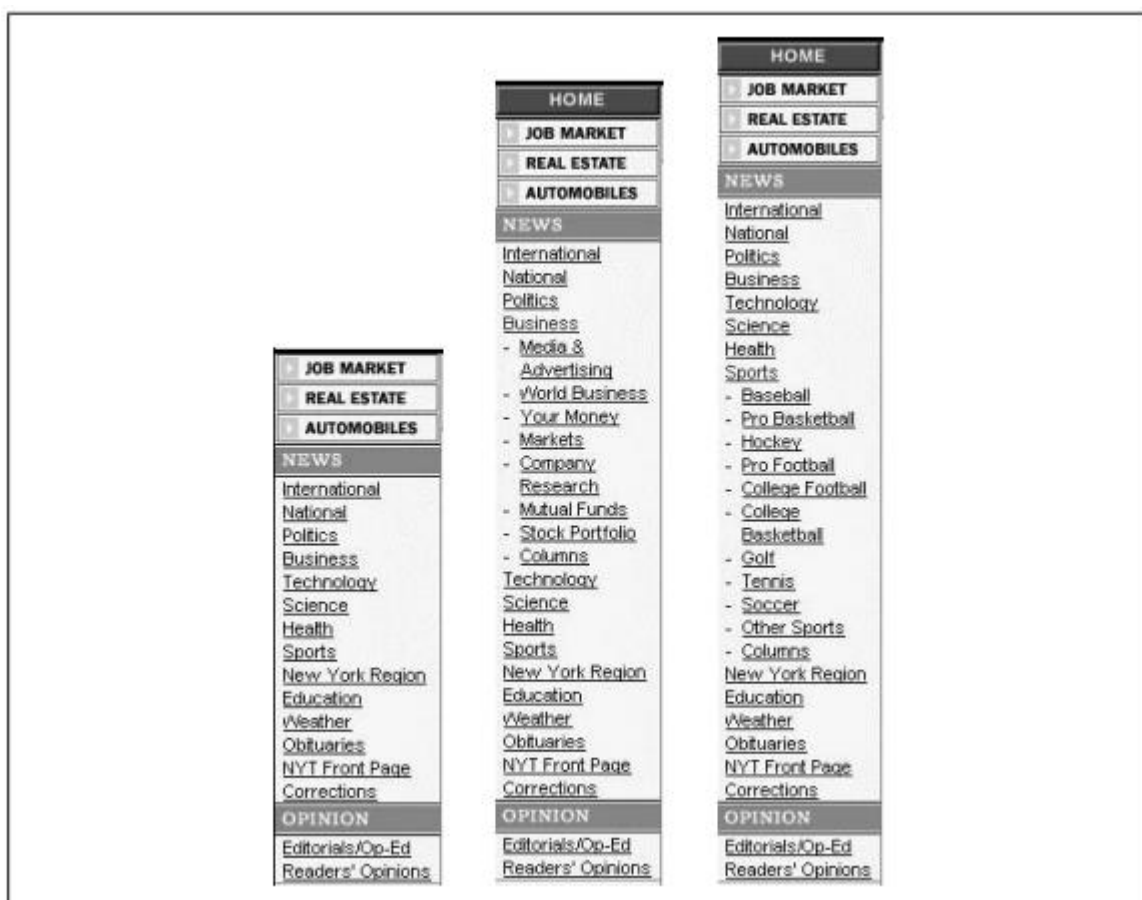
<sup>6</sup>[...] a distinct collection of hypertext links that connect a series of pages, enabling movement among them. They can support global, local, and contextual navigation. You can implement navigation in all sorts of ways, using text or graphics, pull-downs, pop-ups, rollovers, cascading menus, and so on.

<sup>7</sup>Does the local navigation bar work best at the top of the page, or is it better running down the left side? Should we use pull-downs, pop-ups, or cascading menus to reduce the required number of clicks? Will users ever notice gray links? Isn't it better to use the blue/red link color convention?

<sup>8</sup>It is often implemented in the form of a navigation bar at the top of each page. These site-wide navigation systems allow direct access to key areas and functions, no matter where the user travels in the site's hierarchy.

*Navegação Local* é um sistema de navegação que permite aos usuários explorar uma área imediatamente. Algumas vezes encontrado com mais de um em um único *site* e que complementa um sistema de navegação global. “Alguns sites rigorosamente controlados integram a navegação global e local em um sistema consistente e unificado” (MORVILLE; ROSENFELD, 2006, p. 124, tradução nossa<sup>9</sup>). Veja um exemplo de navegação local retirada do livro de Morville e Rosenfeld.

**FIGURA 6** – Exemplo de Local Navigation



Fonte: *Information Architecture for the World Wide Web* (2006, p. 124).

*Navegação Contextual*: As categorias de navegação global e local às vezes não abrangem alguns relacionamentos. Isso demanda que se gere *links* de navegação contextual específicos para uma página.

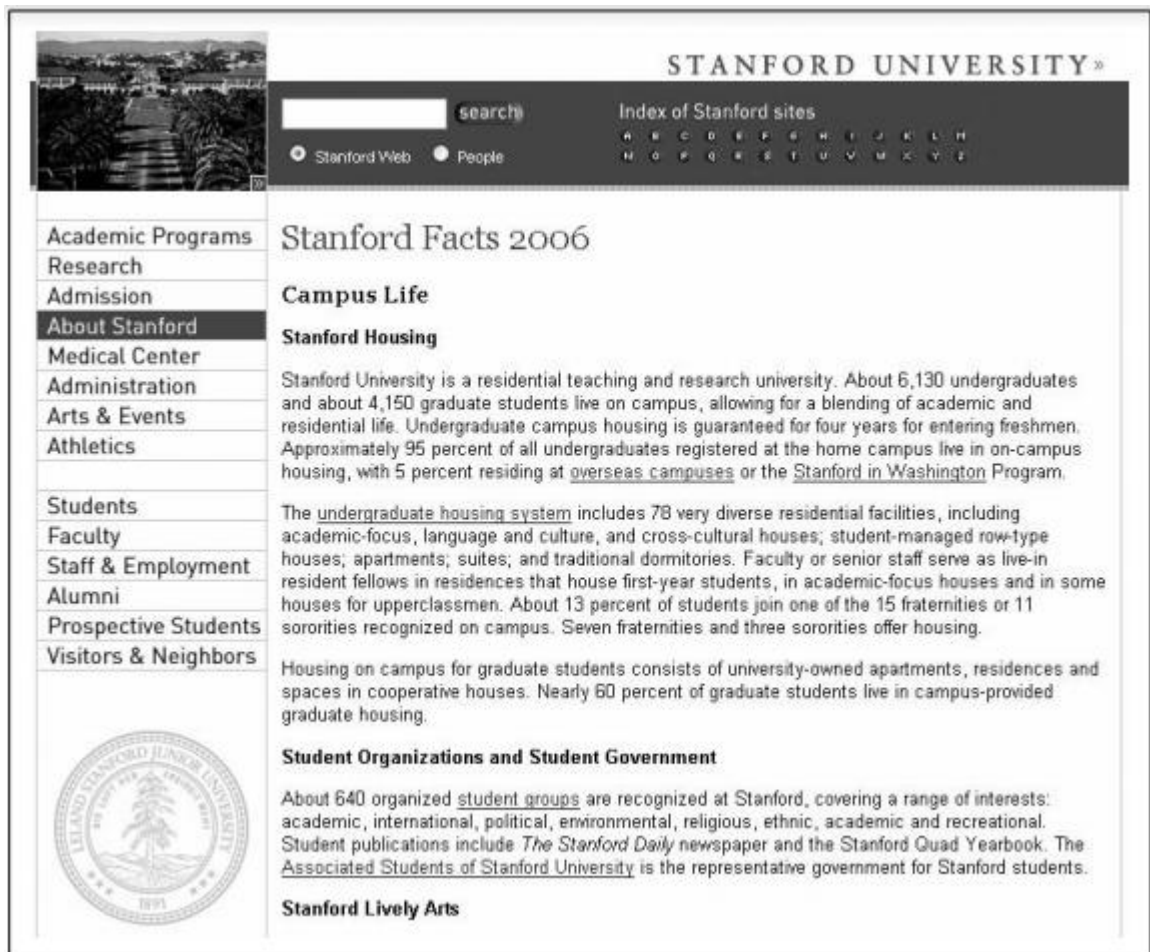
<sup>9</sup> Some tightly controlled sites integrate global and local navigation into a consistent, unified system.



Em um site de comércio eletrônico, esses links "Veja também" podem direcionar os usuários para produtos e serviços relacionados. Em um site educacional, eles podem apontar para artigos semelhantes ou tópicos relacionados (2006, p. 126, tradução nossa<sup>10</sup>).

Veja um exemplo de navegação contextual retirada do livro de Morville e Rosenfeld.

**FIGURA 7** – Exemplo de Contextual Navigation



Fonte: *Information Architecture for the World Wide Web* (2006, p. 126).

Alguns questionamentos sobre a construção da barra de navegação são levantados quanto ao seu *design*. Alguns programadores colocam imagens nos ícones da barra. Realmente a torna mais atrativa. Contudo, deve-se levar em consideração a performance do *site* no momento de carregá-lo. Muitas imagens tornam um *site* mais pesado e usuários portadores de uma *internet* sem fio ou *mobile* serão fortemente prejudicados neste quesito. Sem falar dos usuários cegos que terão que esperar o leitor de tela descrever cada

<sup>10</sup>On an e-commerce site, these “See Also” links can point users to related products and services. On an educational site, they might point to similar articles or related topics.

imagem para ele tomar conhecimento do que se trata aquele item do menu. Barra de navegação com imagem não é nada acessível.

Morville e Rosenfeld (2006) afirmam que, por convenção, a barra de navegação global costuma aparecer no topo do *site*, enquanto a barra de navegação local ao lado esquerdo. Entretanto, há vários *sites* com diferentes posicionamentos de barras de navegação. Apenas é necessário que a usabilidade seja testada caso o desenvolvedor decida não seguir as convenções.

Outro questionamento levantado no livro *Information Architecture on the World Wide Web* é o uso de *labels* ou ícones. As *labels* são etiquetas que podem ser seguidas de uma imagem em seu interior, mas que tem como principal elemento um texto. Elas são mais fáceis de criar e mais transparentes para indicar conteúdo. Os ícones, por outro lado, são mais difíceis de criar e trazem uma forte ambiguidade. Uma imagem sem um texto pode ser interpretada de várias formas, ainda mais em uma barra de navegação com extensões globais.

Na próxima seção, iremos estudar um pouco mais sobre uma das áreas de atuação da AI: a acessibilidade na *web*.

#### **4.2.2. ACESSIBILIDADE WEB**

Acessibilidade é a qualidade atribuída àquilo que é acessível, ou seja, àquilo que é alcançável, àquilo que possui acesso fácil no trato e na aquisição. Nas últimas décadas ela está diretamente ligada às preocupações de arquitetos e urbanistas em prol do fornecimento de ensejos às pessoas portadoras de deficiência ou com mobilidade reduzida para o direito de aproveitamento dos espaços públicos com total segurança e autonomia.

O Decreto Federal nº 5.296/2004, em seu artigo 8º, I, define:

I – acessibilidade: condição para utilização, com segurança e autonomia, total ou assistida, dos espaços, mobiliários e equipamentos urbanos, das edificações, dos serviços de transporte e dos dispositivos, sistemas e meios de comunicação e informação, por pessoa portadora de deficiência ou com mobilidade reduzida.

Essa definição abrange vários objetos que necessitam de acessibilidade desde meios de transporte até meios de comunicação. Apesar de bem relevante, é importante analisarmos nas duas últimas linhas que a acessibilidade não é exclusividade para pessoas portadora de deficiência ou com mobilidade reduzida. Além de que o termo “pessoa portadora de deficiência” é inadequado, segundo a Cartilha Acessibilidade na Web W3C Brasil (2013, p. 20).

A Norma Brasileira ABNT NBR 9050:2004 define em seu item 3.1: “Acessibilidade: Possibilidade e condição de alcance, percepção e entendimento para a utilização com segurança e autonomia de edificações, espaço, mobiliário, equipamento urbano e elementos” (2004, p. 1). A principal contribuição dessa definição, como a Cartilha W3C Brasil menciona, é destacar a relevância dos aspectos de alcance, discernimento e assimilação.

A Cartilha Acessibilidade na Web W3C Brasil define, então, acessibilidade como:

[...] possibilidade e condição de alcance, percepção e entendimento para a utilização, em igualdade de oportunidades, com segurança e autonomia, do meio físico, do transporte, da informação e da comunicação, inclusive dos sistemas e tecnologias de informação e comunicação, bem como de outros serviços e instalações (2013, p. 21).

A discussão a favor das diretrizes sobre a acessibilidade já possui seu espaço no âmbito acadêmico há alguns anos. Dentro de um contexto da sociedade da informação, essa difusão de conhecimento é fortemente benéfica.

A Iniciativa de Acessibilidade na Web, em inglês *Web Accessibility Initiative* (WAI) é um empenho do W3C para melhorar os padrões que visam acessibilidade na *web*. WAI afirma que acessibilidade na *web* significa<sup>11</sup>:

[...] que pessoas com deficiência podem usar a web. Mais especificamente, a acessibilidade na web significa que pessoas com deficiência podem perceber, entender, navegar, interagir e contribuir para a web. Ela também beneficia outras pessoas, incluindo pessoas idosas com capacidades em mudança devido ao envelhecimento (2005, tradução nossa<sup>12</sup>).

A acessibilidade na *internet* é a flexibilização do acesso à *web*, é tornar alcançável a informação nos sites para aqueles que possuem alguma necessidade especial física ou cognitiva no que se refere aos métodos de navegação e de apresentação de *slides*, à operação com *software* e com *hardware* e às adaptações aos ambientes e situações (CONFORTO; SANTAROSA, 2002 *apud* GUIA, 1999). A acessibilidade passa a ser compreendida como um sinônimo de aproximação, um meio de providenciar a cada usuário uma interface que seja capaz de respeitar suas necessidades e preferências.

---

<sup>11</sup> Disponível em: <<https://www.w3.org/WAI/intro/accessibility.php>>. Acesso em: 19 nov. 2017.

<sup>12</sup> Web accessibility means that people with disabilities can use the Web. More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web. Web accessibility also benefits others, including older people with changing abilities due to aging.

Pessoas que não estão familiarizadas com a ausência de acessibilidade na *web* não compreendem que o usuário ao operar as funcionalidades de uma página na *internet* pode ser incapaz de ver, escutar e mover-se por meio do conteúdo, o que acarreta uma absorção de informação ineficiente; possuir um modem mais lento o que inibe a visualização da interface gráfica, utilizar-se de algum equipamento que não possui saída para áudio; não falar ou não compreender a língua em que o documento está escrito; encontrar-se em situações em que seus olhos, ouvidos ou mãos estejam ocupados (CONFORTO; SANTAROSA, 2002).

Com uma *web* acessível, muitos contextos que aparentemente são considerados improváveis se tornariam possíveis, além disso, não contemplariam apenas pessoas com deficiência, mas também para qualquer categoria de usuário (Cartilha Acessibilidade W3C Brasil, 2013, p. 29):

- i. uma mulher cega, utilizando um leitor de telas, pesquisa a restituição de imposto de renda no sítio da Receita Federal;
- ii. um homem cego e sem braços procura sua ex-professora em um sistema de busca utilizando um programa de reconhecimento de voz para entrar comandos no computador e receber retorno a partir do leitor de telas;
- iii. um homem com paralisia cerebral, com grandes dificuldades motoras e que só utiliza um dedo para teclar, atualiza seu perfil em uma rede social;
- iv. um homem com deficiência motora, que usa um mouse adaptado, faz compras em uma loja virtual;
- v. uma jovem tetraplégica, utilizando apenas um ponteiro na cabeça, procura informações sobre células-tronco em sítios especializados;
- vi. uma mulher com baixa visão procura informações sobre investimentos e a crise econômica mundial, utilizando um programa ampliador de tela;
- vii. um programador daltônico testa uma aplicação na web, procurando erros;
- viii. um jovem surdo ou com deficiência auditiva que faz um curso de inglês a distância.

Como mencionado pela cartilha, essas situações abrangem qualquer categoria de usuário e isso inclui situações como um estrangeiro acessar o *site* de um outro país para fins de intercâmbio pessoal, por exemplo. O multilinguismo é uma forma extraordinária de difusão de informação. A fim de conceber o devido esclarecimento de sua importância para a acessibilidade *web*, o separamos em uma nova sessão que veremos a seguir.

#### 4.2.2.1. MULTILINGUISMO

O multilinguismo é caracterizado de maneira simples como o uso de múltiplas línguas. Como muitas áreas vão tomando mais espaço no ciberespaço conforme o avanço da tecnologia na sociedade, o multilinguismo não seria diferente. Segundo Francisco Cláudio Sampaio de Menezes, mestre em Engenharia de Sistemas e Computação pelo Instituto Alberto Luiz Coimbra de Pós Graduação e Pesquisa de Engenharia, afirma, em *O multilinguismo e as novas tecnologias das Línguas no século XXI*, que em prol da permanência e garantia do uso de cada língua nos dias de hoje, autoridades e especialistas começaram a se preocupar com tal assunto por conta da presença das línguas na *internet*.

A preservação do multilinguismo na *internet* é de extrema relevância em prol da acessibilidade da informação e difusão de conhecimento. Por meio de um ambiente multilíngue, é possível aproximar e conectar pessoas que fisicamente estão além das fronteiras de um país. Páginas de *web* multilíngue proporcionarão a possibilidade de que uma pessoa da América do Norte possa alcançar o mesmo conhecimento que uma pessoa da Ásia, por exemplo. Menezes afirma:

Considerando, por exemplo, o recente crescimento dos blogs, tecnologia que permite a publicação pessoal, como alcançar audiência de falantes de outras línguas? Como esse interesse em alcançar novas audiências pode afetar a escolha da língua de publicação nos blogs? Como assegurar a interoperabilidade das aplicações? Como alcançar a clientela de outros países por meio do comércio(*sic*) eletrônico? (2015, p. 95)

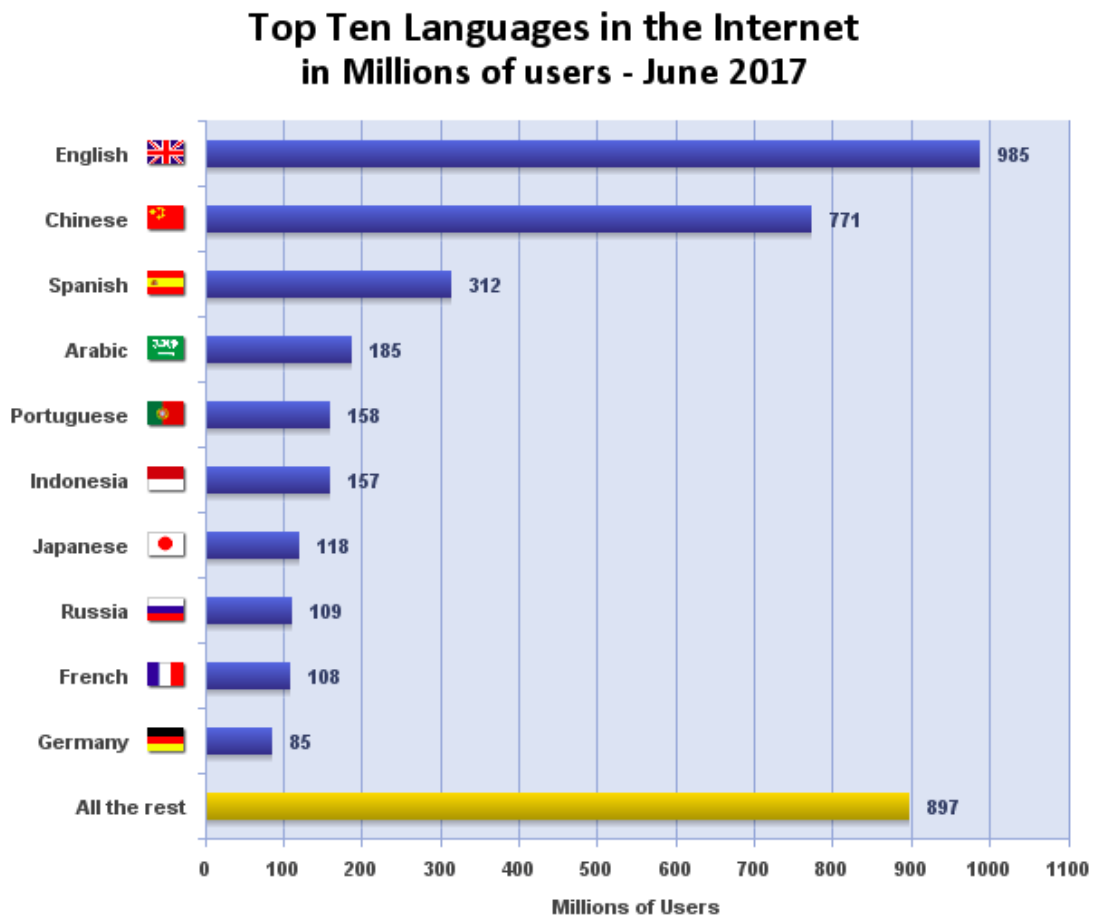
As questões levantadas pelo o autor são relevantes para o ponto de vista do multilinguismo como catalisador de usuários no que diz respeito ao acesso de uma página na *web*. É muito corriqueiro ver pessoas escrevendo conteúdo de seu *site* pessoal ou até mesmo comercial em inglês – que é a língua mais difundida no mundo – com a finalidade de expandir o número de usuários. Do contrário, escrevendo apenas em sua língua nativa, pode acabar restringindo ao acesso apenas de pessoas do seu país.

O gráfico do *Internet World Stats*<sup>13</sup> na Figura 8 mostra as dez línguas mais presentes na internet em 2017 e a partir dele podemos ter uma noção do quão benéfico é o uso do multilinguismo em *sites*. Se um nativo francófono desenvolvesse um *site* com, além do francês, mais os três primeiros idiomas indicados no gráfico, ele aumentaria sua possibilidade de acesso de 78.9 milhões para mais de um bilhão de possíveis acesso – claro, levando em consideração o teor e conteúdo de seu *site*, que pode ser diverso.

---

<sup>13</sup> Disponível em: < <http://www.internetworldstats.com/stats7.htm>>. Acesso em: 18 nov. 2017.

FIGURA 8 – Top dez línguas na internet em 2017



Fonte: *Internet World Stats*.

O multilinguismo estará no *corpus* de estudo deste projeto visando o desenvolvimento de um *site* acessível.

No próximo capítulo, iremos analisar alguns *sites* e veremos quais são as *features* mais utilizadas para suprir a carência de acessibilidade e usaremos o resultado como parâmetro para construção da nossa lista de *features* de acessibilidade a serem estudadas e desenvolvidas nos próximos capítulos.

## 5. SITES CORRELATOS

Ao final deste capítulo, iremos listar as *features* de acessibilidade mais utilizadas e encontradas na *web*, para isso foi realizada uma pesquisa em *sites* governamentais e outros com a acessibilidade como principal conteúdo.

O Acessibilidade Legal<sup>14</sup> é um *site* desenvolvido por Marco Antônio de Queiroz, um programador cego que já deu diversas palestras sobre acessibilidade, navegabilidade e arquitetura da informação. Sua página na *web* é acessível para pessoas cegas e de baixa visão e pessoas com problemas motores que enfrentam dificuldade para navegar com o cursor.

**FIGURA 9** – Cabeçalho da página do site do Acessibilidade Legal



Fonte: Elaborada pelo autor (captura de tela).

O *site* Acesso Para Todos<sup>15</sup> também contém ferramentas para saltar ou para o conteúdo ou para o menu e aumento e diminuição de fonte. Um grande diferencial deste *site* é que ele possui o aplicativo (HandTalk) para LIBRAS no canto direito.

**FIGURA 10** – Cabeçalho da página do site do Acesso Para Todos



Fonte: Elaborada pelo autor (captura de tela).

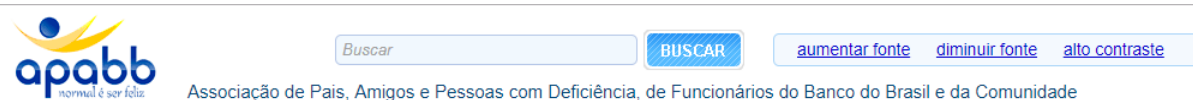
O *site* Associação de Pais, Amigos e Pessoas com Deficiência, de Funcionários do Banco do Brasil e da Comunidade (APABB)<sup>16</sup> também possui algumas *features* de acessibilidade.

<sup>14</sup> Disponível em: <<http://acessibilidadelegal.com/>>. Acesso em: 30 out. 2017.

<sup>15</sup> Disponível em: <<https://www.acessoparatodos.com.br/>>. Acesso em: 30 out. 2017.

<sup>16</sup> Disponível em: <<http://www.apabb.org.br/>>. Acesso em: 19 nov. 2017.

**FIGURA 11** – Cabeçalho da página do site da APABB



Fonte: Elaborada pelo autor (captura de tela).

É interessante perceber que todos os *sites* possuem uma estrutura mínima de acessibilidade: pular para o conteúdo, pular para o menu, aumentar e diminuir fonte e contraste. A maioria destas ferramentas são apenas para usuários cegos. O *site* Vida Mais Livre<sup>17</sup> é outro exemplo com uma estrutura mínima de acessibilidade.

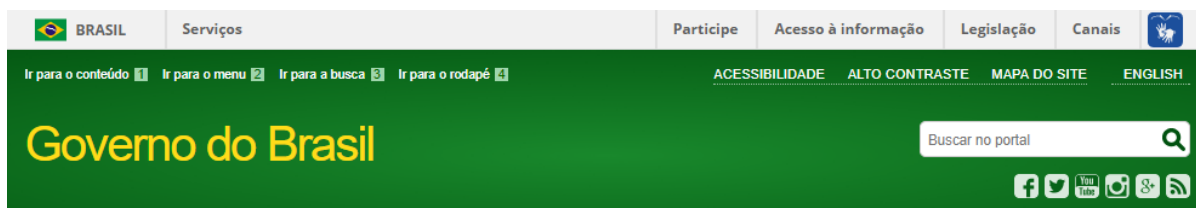
**FIGURA 12** – Cabeçalho da página do site da Vida Mais Livre



Fonte: Elaborada pelo autor (captura de tela).

No próximo exemplo, teremos um *site* com uma estrutura acessível um pouco mais completa porque além da acessibilidade para cegos e para surdos, ele é provido da acessibilidade multilingue. O *site* do Governo do Brasil<sup>18</sup> é um exemplo de usabilidade e navegabilidade para uma gama enorme de usuários. Sua informação atinge um número de usuários muito maior em comparação aos outros mencionados neste capítulo porque ele contém a *feature* de multilinguismo. O uso desta ferramenta não deveria ser restringido apenas a *sites* governamentais.

**FIGURA 13** – Cabeçalho da página site do Governo do Brasil



Fonte: Elaborada pelo autor (captura de tela).

<sup>17</sup> Disponível em: <<https://vidamaislivre.com.br/>>. Acesso em: 30 out. 2017.

<sup>18</sup> Disponível em: <<http://www.brasil.gov.br/>>. Acesso em: 19 nov. 2017.



Mesmo sabendo da importância de formas de acessibilidade para usuários com necessidades especiais motora ou cognitiva por exemplo, existe uma estrutura mínima de acessibilidade para *web* que abrange majoritariamente cegos e surdos. Com base na análise dos *sites* apresentados neste capítulo e a explicitação de algumas práticas em prol da acessibilidade *web* e de alguns exemplos em certos *sites*, neste trabalho iremos trabalhar com as seguintes *features* de acessibilidade:

- Aumento e diminuição de fonte para pessoas com baixa visão;
- Contraste de cores de fundo para pessoas com baixa visão;
- Pular para o conteúdo para pessoas cegas;
- Destacar e sublinhar *links* para pessoas com baixa visão;
- Janela virtual de tradução de LIBRAS para pessoas surdas e ensurdecidas;
- Multilinguismo para pessoas estrangeiras.

É indiscutível que exista muitas outras formas de transformar um *site* acessível para diversas outras necessidades especiais além da cegueira e da surdez. Todas merecem a devida atenção, porém, para o objetivo deste trabalho, iremos trabalhar apenas com aquelas mencionadas na lista. No próximo capítulos veremos um pouco mais algumas boas práticas com HTML em prol da acessibilidade.

## **6. BOAS PRÁTICAS DE ACESSIBILIDADE NA WEB**

Assim como há protocolos e regras a serem seguidas para acessibilidade para espaços públicos e sociais, como a altura e comprimento de uma rampa ou a largura de um piso tátil, a acessibilidade na *web* também requer boas práticas provindas de estudos em prol de uma navegabilidade e usabilidade efetiva.

No site oficial da W3C é possível encontrar muito conteúdo do que é necessário que uma página *web* tenha para ser considerada acessível. É possível encontrar material que informa como tornar *sites* acessíveis para todos os tipos de usuários, não necessariamente para aqueles portadores de alguma deficiência, mas também usuários videntes e ouvintes ou que se encontram em situações diversas e que precisam ter uma navegabilidade limpa e intuitiva e que não encontrem algum desconforto para encontrar alguma informação. Então, neste capítulo, iremos conhecer algumas dessas práticas que favorecem o manuseio e a experiência do usuário na *web* e no final deste, iremos definir com quais *features* de acessibilidade iremos trabalhar e ensinar como desenvolvê-las em um *site*.

Primeiro, vamos lembrar o que é CSS e HTML. *Hypertext Markup Language* (HTML), que em português é Linguagem de Marcação de Hipertexto, é uma linguagem de marcação que estrutura e organiza conteúdos em páginas da *web* por meio de componentes chamados de *tags*. As tags são palavras entre parênteses angulares (chevron) < e > e são elas que dão os comandos de formatação do código. *Cascading Style Sheet* (CSS) é um mecanismo de linguagem que gerencia o estilo (cor, tamanho, espaçamento etc) de uma página *web* escrita em HTML. Segundo a publicação de Youseph Santos, Engenheiro *Front-End* da Totvs, no *site* Maujor<sup>19</sup> sobre CSS e padrões para *web*, ao estruturar o conteúdo do *site*, o HTML visa dois grandes objetivos:

traduzir os conteúdos que o ser humano interpreta facilmente para uma linguagem que os meios de acesso a Web entendam e exibir os conteúdos de forma organizada. Sendo assim, um código semântico torna a navegação mais acessível, incluindo portadores de necessidades especiais, uma vez que os meios de acesso entendem o conteúdo e podem passar para o usuário com facilidade (SANTOS, 2016).

Como o engenheiro afirma, é muito importante a semântica do código estar bem estruturada a fim de proporcionar uma boa acessibilidade. Com a atualização do HTML5 e suas novas tags é possível trazer novos significados semânticos para o código. O atributo ALT, por exemplo, é uma excelente forma de prover uma alternativa para a visualização de texto. Ao inserir a tag <img>, adicione o atributo ALT para digitar a descrição da imagem. Por algum problema que a imagem possa não estar visível para o usuário, a descrição irá aparecer no lugar da imagem. Esse atributo é fundamental, também, para leitores de tela. Ao passar o cursor do mouse em cima de uma imagem, um usuário cego terá uma audiodescrição pelo leitor de tela.

O atributo TITLE é muito útil para *tooltips* – uma moldura em *pop up* que se abre quando passa o cursor do mouse por cima de algum elemento, como link, por exemplo. Os leitores de tela também identificam essa informação para o usuário e em prol da usabilidade, é uma ferramenta muito útil para agregar uma informação a mais para um *link*.

O atributo LANG é muito utilizado para indicar qual idioma o documento está sendo escrito. Mais uma vez, é muito útil para leitores de tela que possuem capacidade leitora multilíngue. O *NonVisual Desktop Access*<sup>20</sup> (NVDA) é um bom exemplo de leitor de tela. É um programa que lê o conteúdo do *site*, possibilitando que usuários cegos possam usar o

---

<sup>19</sup> Disponível em: <<http://maujor.com/tutorial/escrevendo-html-semantico.php>>. Acesso em: 12 nov. 2017.

<sup>20</sup> Disponível em: <<https://www.nvaccess.org/>>. Acesso em: 18 nov. 2017.

computador. É possível selecionar o conteúdo com o cursor do mouse ou pela tecla TAB do teclado.

Com o HTML5 é possível fazer uso de novos *inputs types* para formulário. Ao usar uma tag <tel> num formulário <form> em um dispositivo móvel, o sistema irá automaticamente disponibilizar acesso ao teclado numérico do dispositivo.

A tecla TAB é a principal tecla quando precisamos navegar pelo teclado. Dependendo do *site*, quando precisamos navegar utilizando a tecla, ela nem sempre foca nos elementos importantes. O atributo TABINDEX serve para resolver isso. Com ele, determinamos o caminho que o foco deve percorrer ao pressionar a tecla TAB.

O W3C não menciona apenas atributos de HTML, mas também como boas práticas de posicionamento de conteúdo em prol da acessibilidade do usuário como a construção de um menu de atalhos. Nem sempre a navegação pelo teclado é confortável. É possível que o usuário tenha que apertar a tecla TAB mais de 30 vezes para acessar o conteúdo do *site*. Por isso, podemos construir um menu simples que será a primeira coisa que o usuário e/ou o leitor de tela terá acesso para ler. Elementos como pular para o conteúdo principal, ou para o *sidebar*, ou para o rodapé, bem como botões de aumento e diminuição de fonte são extremamente relevantes para um menu de atalhos.

Alguns textos e termos do *site* precisam possuir o entendimento o mais claro possível evitando gírias e abreviações a fim de trazer mais detalhes ao usuário. Evita-se a utilização de termos como “SAIBA MAIS” ou “CLIQUE AQUI” em *links* ao final de textos, por exemplo, porque há uma grande possibilidade de o usuário cego não entender quando o leitor de tela ler um texto que termine com tais termos por poder achar que “SAIBA MAIS” faça parte do corpo do conteúdo do texto principal. Nos botões de aumento e diminuição de tela, não abrevie com “A +” e “A -”, porque mesmo que pessoas cegas não utilizarão essa ferramenta por não possuir nenhuma visão, eles irão saber do que se trata e poderá ser evitado uma confusão de informação no menu de atalhos.

Mesmo que este código semântico do HTML ajude a estruturar o *site*, não há uma lista definida com exatamente todas as *features* que um *site* deve conter para ser considerado acessível. Como vimos, há algumas convenções como *World Wide Web Consortium* e *Web Accessibility Initiative* (WAI) que procuram organizar e difundir seu conhecimento a fim de tornar a área mais padronizada. Afinal, a acessibilidade *web* é muito mais que atender ao público cego com um atributo ALT em todas as imagens. Há muitas pessoas com outras formas de necessidades especiais que devemos levar em consideração

como a surdez, a cognitiva, a motora e até mesmo a surdo-cegueira que é uma pessoa surda e cega ao mesmo tempo.

Contudo, antes que comecemos a fazer um estudo sobre como desenvolver as ferramentas de acessibilidade na *web*, é muito importante que façamos um prévio planejamento do nosso *site*. Toda página na *web* precisa ser organizada e estruturada antes que comece qualquer linha de código. De nada valerá as *features* de acessibilidade no *site* se sua arquitetura da informação está totalmente confusa. Planejar a experiência do usuário é tão importante quanto o próprio desenvolvimento, afirmam Ethan Watrall e Jeff Siarto, autores do livro *Head First Web Design* (2009).

### **6.1. LEVANTAMENTO DE REQUISITOS E FLUXOGRAMA**

Para a construção de um *site* é muito importante que se leve em consideração algumas regras e protocolos a serem seguidos para a boa navegabilidade e usabilidade do usuário. Neste ponto do projeto é crucial que entendamos que quando desenvolvemos uma página de *web*, na sua grande maioria, ela deve estar ao alcance dos usuários, ou seja, a não ser que seu *site* tenha única e exclusivamente um propósito pessoal, ele deve seguir algumas boas condutas desenvolvidas ao longo dos anos de estudo de *web design* para que o indivíduo não se sinta perdido ao navegar por sua página quando buscar alguma informação, do contrário, ele irá se entediar, fechar a janela e iniciará uma nova busca no *browser* e poderá encontrar o que procura em outro lugar. Para um *site e-commerce*, neste processo todo, você acabou de perder um cliente em potencial.

Quando falamos usabilidade na *web*, estamos nos referindo sem nenhuma ambiguidade a acessibilidade. Melhorar o *layout* de um *site* para que um vidente não se sinta perdido e confuso em sua busca é o mesmo que inserir um botão de pular para o conteúdo para uma pessoa cega, afinal, esta última já tomou conhecimento de sua barra de navegação e não precisa perder tempo esperando que o leitor de tela a releia tudo para que, enfim, alcance o objetivo de sua busca que era apenas o conteúdo daquela página.

Uma das fases de construção do *layout* é feita pela a inserção de botões e opções que visam o funcionamento da acessibilidade na navegabilidade do usuário e para que isso seja realizado com sucesso, o desenvolvimento deve organizar-se por meio de um fluxograma. Um fluxograma é uma representação gráfica de vários processos ou fluxos de trabalho. Geralmente representado com figuras geométricas e com setas que interligam com outras figuras geométricas para representar o passo a passo a seguir a fim de alcançar um objetivo de trabalho.

Todo e qualquer *site* necessita ser idealizado dentro de um fluxograma. É ele quem irá representar as ideias organizadas dos componentes que devem estar no *site* e isso inclui a acessibilidade. Segundo Ethan Watrall e Jeff Siarto (2009), em *Head First Web Design*, um processo de *design* estrutura seu projeto para que se mantenha firme e não tome diferentes direções ao longo do desenvolvimento. É de extrema importância a organização de ideias dentro de um fluxograma antes de escrever uma única linha de código. Desta forma, não correrá o risco de perder tempo com o montante de ideias que surgem no momento da codificação e assim ter que apagar e começar tudo outra vez.

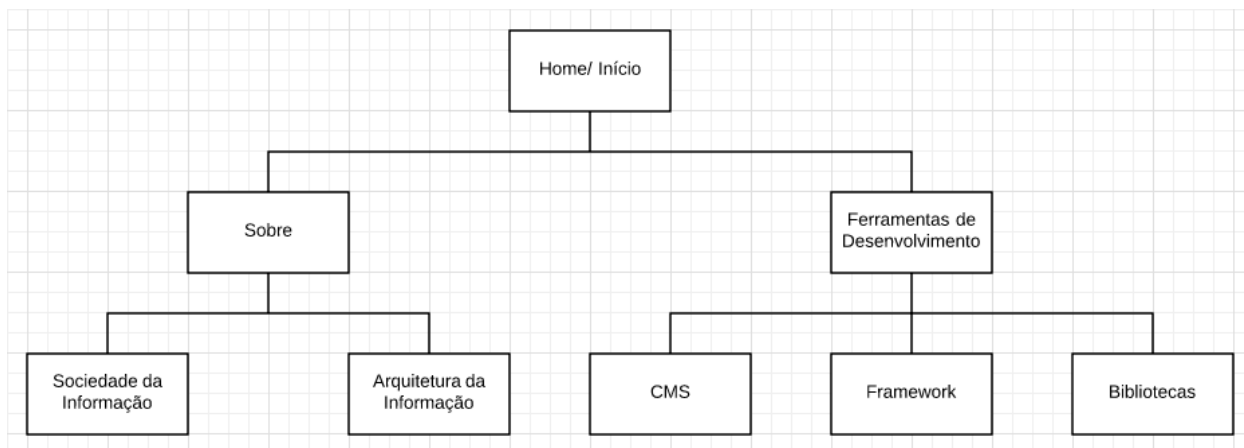
Construir um *site* pode parecer com o processo de construção de uma casa: se começar com a planta baixa planejada, saberá cada passo que terá que tomar para o desenvolvimento. Esse processo na *web* se chama Arquitetura da Informação (AI). Como já vimos, a AI é o processo pelo qual se divide o conteúdo do *site* em pequenas partes e então as organiza hierarquicamente em uma forma lógica. Na maioria do tempo, cada pequena parte de informação é conteúdo de imagem ou texto em uma única página. AI é também muito próxima do conceito de construir a navegabilidade do *site* que essa, por sua vez, está muito próxima de acessibilidade *web*. Logo, se construir uma péssima Arquitetura de Informação, terá uma péssima navegação. Se o *site* não possuir uma sólida AI, ele será desorganizado e confuso para os usuários e isso fará com que eles vão para qualquer outra página procurar o que estão procurando.

### 6.1.1. CONSTRUINDO UM FLUXOGRAMA

Para melhor explicação deste processo nesta seção, iremos simular a solicitação de construção de um *site* por um cliente. Vamos chamá-lo de João. João quer um *site* simples. A *home* precisa estar estruturada em estilo *one-page* – quando várias ou todas informações estão em uma única página e o usuário apenas usará a rolagem para navegação do *site*, desta forma o usuário não precisará clicar em vários *links* para obter as informações. O *site* não precisa de uma sessão de contato, porque João não tem fins comerciais, mas definitivamente precisa de uma sessão sobre o seu trabalho desenvolvido no *site* e do que se trata seu projeto. Por fim, João também lembrou de que o *site* precisa ser multilíngue a fim de permitir o acesso de seu trabalho a estrangeiros.

De acordo com os pedidos de João, o fluxograma de seu *site* ficaria de acordo com a imagem abaixo:

**FIGURA 14** – Fluxograma do site



Fonte: Elaborada pelo autor (captura de tela).

Assim que o usuário entrar no início da página, ela terá as opções em *dropdown* na *navbar* de “Sobre” e “Ferramentas”. Isso melhorará sua navegabilidade para que não haja tanta informação em linha reta. Nas opções de *dropdown* de “Sobre” terão “Sociedade da Informação” e “Arquitetura da Informação”. É nesta sessão que o *site* do João informará de forma geral do que se trata sua página *web*. Nas opções de *dropdown* “Ferramentas” terão “CMS”, “Framework” e “Bibliotecas”. É nesta sessão que haverá a informação do trabalho desenvolvido pelo João.

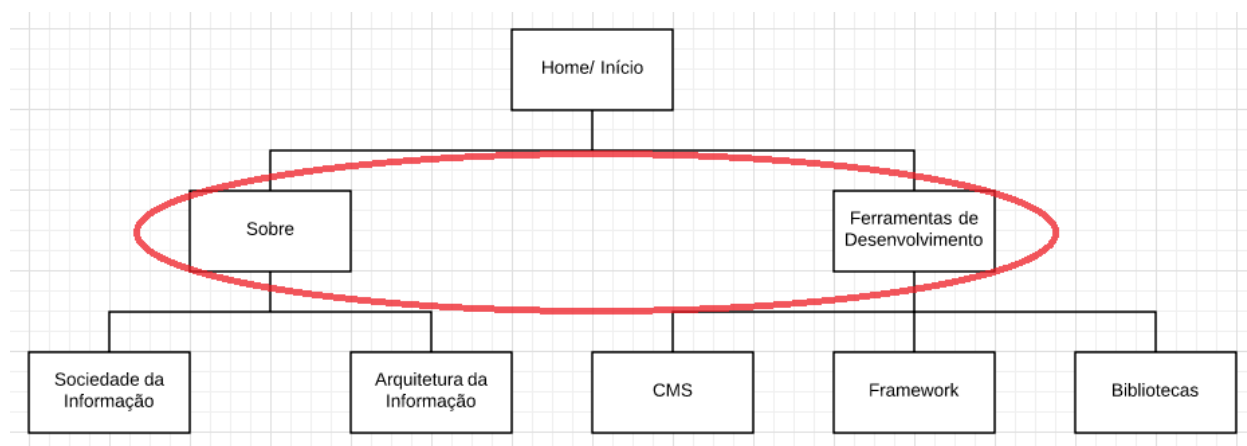
Iniciando qualquer página na *web* com fluxograma organizado, mesmo que simples, a equipe de desenvolvimento não estará propícia a esquecer das boas práticas da acessibilidade. Desde o início, eles saberão onde é melhor inserir os botões e ferramentas de acessibilidade para usuários com alguma necessidade especial física ou cognitiva. Aqui não esquecemos dos usuários estrangeiros. Tornar um *site* multilíngue também faz parte de acessibilidade. Ela não está presente no fluxograma porque esta ferramenta mostra apenas a navegabilidade do usuário entre as páginas e não botões. Veremos sobre *layout*, cores e posicionamento das informações e botões mais a frente quando desenvolvermos o esboço.

Após estruturar o fluxograma, segundo Watrall e Siarto (2009), é necessário determinar qual será sua *top navigation*. A navegabilidade no topo da página geralmente é o elemento mais proeminente: a barra de navegação com abas na parte superior da página ou o menu de navegação vertical em uma coluna secundária. Frequentemente, essas abas se conectam a sessões um nível mais abaixo, representado também no fluxograma.

Dentro da nossa planta, identificamos a *top navigation* com o círculo na Figura 15. É um tanto quanto intuitivo identificar essa aba de navegabilidade no fluxograma por causa da presença do elemento *Home/Início*, uma vez que é o elemento de início da

navegação, o menu que estiver próximo a ele com os macroelementos será considerado a *top navigation* (WATRALL; SIARTO, 2009).

**FIGURA 15** – Identificação da Top Navigation no fluxograma do site

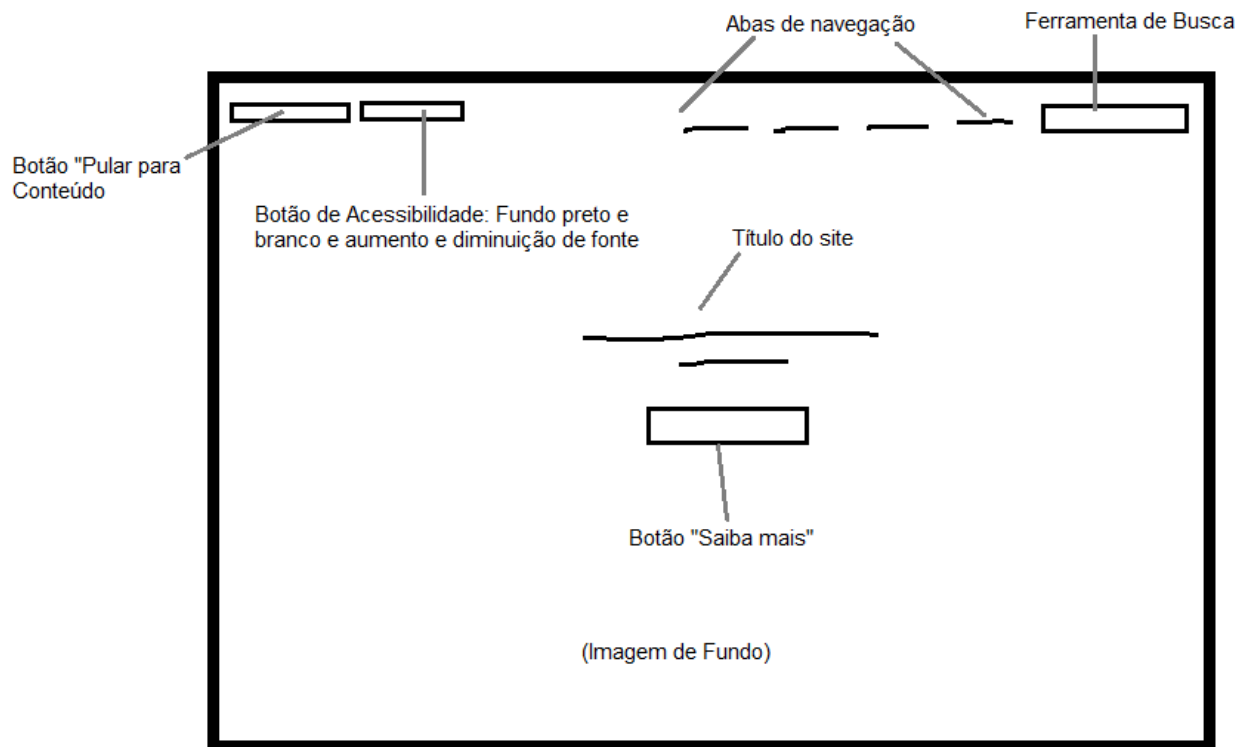


Fonte: Elaborada pelo autor (captura de tela).

Uma vez a *top navigation* identificada, é fácil analisarmos onde colocar os botões e ferramentas de acessibilidade para usuários que a necessitam. A identificação das abas de navegação em prol de uma boa usabilidade e capacidade de busca não favorece apenas os usuários videntes. Usuários com necessidades especiais merecem a mesma atenção no quesito experiência do usuário e nessa fase de estruturação de planta os desenvolvedores devem redobrar a atenção.

Na próxima etapa do desenvolvimento devemos desenhar o esboço do *site*. Nessa etapa teremos a possibilidade de ter uma noção espacial do site. O quanto algum componente de texto e/ou imagem ocupará de espaço físico na página. Onde iremos alocar as abas de navegabilidade e alguns botões importantes, como por exemplo as ferramentas de acessibilidade. Watrall e Siarto (2009) afirmam que o esboço de uma página *web* mantém o foco na funcionalidade. Ao invés de perder tempo codificando sem nenhum planejamento o *background* ou o *font-size* que provavelmente seria mudado posteriormente, o esboço auxilia no desenvolvimento da arquitetura da informação bem como a experiência do usuário e mais uma vez, isso contempla a acessibilidade. O esboço pode e é recomendável ser feito a mão num papel, para que quando desenvolvido a pedido de um cliente, este se sinta mais familiarizado com o processo. Nem todos lidam bem com as tecnologias. Abaixo, veja a imagem do esboço do início do *site* exemplo que será desenvolvido com *framework* e biblioteca.

**FIGURA 16** – Esboço do site



Fonte: Elaborada pelo autor (captura de tela).

A vantagem de desenvolver o esboço é que ele nos dá parâmetros visuais que não teríamos apenas com o fluxograma. Logo, após o seu desenvolvimento, poderíamos implementar as cores e componentes mais sólidos na planta da página *web*. Mesclar as cores neste momento é crucial para a experiência do usuário daltônico. Adicionar cores para o esboço fornece a chance de visualizar como as cores interagem uma com as outras, bem como a interface e os elementos do *layout* se comportam um com o outro quando eles estão coloridos.

### 6.1.2. A PROGRAMAÇÃO E ALGUNS CONCEITOS

Agora que João já tem todo seu *site* estruturalizado, é hora de programar. Nesta etapa final, precisamos ter conhecimento de algumas formas de construir um *site*. Segundo o *site* O'Reilly<sup>21</sup> (2017), existem 2500 linguagens de programação documentadas. E por que será que existem tantas? Para cada linguagem, há uma solução de algum problema. É normal que na área de desenvolvimento de *software* e sistemas encontrem alguns problemas a serem resolvidos e cada linguagem de programação contempla essas soluções.

<sup>21</sup> Disponível em: <[http://archive.oreilly.com/pub/a/oreilly/news/languageposter\\_0504.html](http://archive.oreilly.com/pub/a/oreilly/news/languageposter_0504.html)>. Acesso em: 25 out. 2017.



Com a existência de tantas linguagens de programação hoje em dia, é possível encontrar outras formas de construir *sites* sem programar uma linha de código, bem como há certas ferramentas previamente já desenvolvidas que facilitam o desenvolvimento caso construa um *site* programando e outras ferramentas acabam dinamizando absurdamente o trabalho do programador atingindo funcionalidades autossuficientes e automatizadas.

Nos capítulos 7, 8 e 9 iremos, então, fazer o estudo e o desenvolvimento das ferramentas que viabilizam a lista de *features* criada no capítulo 5. Sites correlatos, seguindo como métrica o nível de complexidade. O capítulo 7 abordará o nível básico com o Sistema de Gerenciamento de Conteúdo, onde o desenvolvedor fará uso apenas de arrastar e soltar *templates* prontos e talvez usar alguns *plug-ins*, os quais fornecerão toda a funcionalidade do *site* automaticamente. São frequentemente usados por pessoas com pouco ou nenhum conhecimento de programação; o capítulo 8 abordará o nível intermediário com o *framework* onde o desenvolvedor já deve ter um conhecimento prévio de HTML e CSS para usar um *framework* a fim de possuir alguns componentes já prontos para dinamizar e facilitar a construção do *site*, mesmo que requeira um pouco de programação; e o capítulo 9 abordará o nível avançado com a biblioteca onde o desenvolvedor já deve ter um conhecimento intermediário de HTML e CSS para saber manusear bibliotecas prontas a fim de alcançar um nível de *design* e *layout* mais complexo. Cada uma dessas ferramentas possui seus pontos fortes e fracos que serão decorridos ao longo de seus respectivos capítulos.

## **7. SISTEMA DE GERENCIAMENTO DE CONTEÚDO (CMS)**

Um Sistema de Gerenciamento de Conteúdo, em inglês *Content Management System* (CMS), é uma aplicação de computador que é capaz de criar e modificar conteúdo e componente de um *site*. Uma de suas vantagens é que há uma grande variedade de recursos para CMS. Geralmente ele é gerenciado por vários usuários que trabalham em conjunto em um ambiente corporativo. A maioria dos CMSs inclui publicação baseada na Web, gerenciamento de formatos, edição de histórico e controle de versão, indexação, pesquisa e recuperação.

Segundo Fernando Chagas (*apud* BOIKO, 2001), em *Um estudo sobre os sistemas de gerenciamento de conteúdo de código aberto*,

Um SGC possibilita a criação, o gerenciamento, a distribuição, a publicação e a recuperação de informações corporativas, sendo que o gerenciamento do conteúdo deve ser definido sob o ponto de vista das atividades das pessoas e dos seus objetivos [8]. Qualquer colaborador de uma organização, detentor de informação, pode produzir conteúdo no website da organização com o uso

de um SGC. Além disso, os SGC reduzem erros de publicação e facilitam o processo de validação (2008, p. 1).

O sistema mais popular hoje é *WordPress* e será com ele que iremos trabalhar. Decidimos trabalhar com *WordPress* por ser a plataforma mais usada na *internet*. Um a cada quatro *sites* encontrados na *web* é feito pela plataforma. Nessa escala, 25% das páginas presentes na rede pertence ao sistema (TECMUNDO, 2015). Essa informação é de extrema relevância para a seleção do *corpus* deste trabalho uma vez que o intuito é sempre buscar ferramentas em prol da difusão da informação por meio da acessibilidade *web*.

## 7.1. WORDPRESS

O WordPress é um sistema de gerenciamento de conteúdo gratuito e de código aberto baseado em PHP (linguagem de programação com um *script* de código aberto muito utilizada para desenvolvimento *web* e de fácil integração com HTML) e MySQL (sistema gerenciador de banco de dados relacional de código aberto muito utilizado em aplicações por ser gratuito). Existem duas opções para fazer com que o CMS funcione: a primeira seria hospedando em seu próprio servidor *web* chamado de WordPress.com<sup>22</sup> que é uma hospedagem desenvolvida pelos próprios criadores e a segunda seria parte de um serviço de hospedagem na internet para executar o pacote de *software* WordPress.org<sup>23</sup>. Os recursos do CMS incluem uma arquitetura de *plug-in* e um sistema de modelo. O WordPress, além de ter sido usado por mais de 27,5% dos 10 milhões de *sites* mais importantes desde fevereiro de 2017, é o sistema de blogs mais popular em uso na *web*, abrangendo mais de 60 milhões de sites segundo seu *site* brasileiro.

Apesar algumas desvantagens do CMS, como a baixa segurança por ser muito popular, a constante necessidade de atualização dos *plug-ins* exatamente em prol da segurança, além do fato de que as ferramentas com funcionalidades mais complexas são pagas em dólares, alguns dos motivos que nos encorajou fortemente a escolher o WordPress é porque a plataforma é fácil de instalar e gerenciar, o que se torna acessível para pessoas mais leigas com a programação. Ele possui dezenas de temas disponíveis, o que agrada fortemente os usuários, é altamente versátil, possui fácil otimização para mecanismos de busca, fácil otimização para dispositivos móveis, muita informação disponível e a plataforma pode ser instalada gratuitamente.

---

<sup>22</sup> Disponível em: <<https://wordpress.com/>>. Acesso em: 20 nov. 2017.

<sup>23</sup> Disponível em: <<https://wordpress.org/>>. Acesso em: 20 nov. 2017.

Para começar a desenvolver um *site* pelo WordPress, primeiro é importante decidir se irá hospedá-lo em algum *host* fornecido por terceiros (WordPress.org) ou se irá usar o próprio servidor da plataforma (WordPress.com). As vantagens do WordPress.org sobre o WordPress.com é que no primeiro você tem mais ferramentas e funcionalidades. Ele é mais robusto e lhe dá um controle maior de detalhes e dispositivos. A desvantagem sobre seu colega é que o usuário precisará arcar com os custos e manutenção de um servidor para criar o *site*. Neste projeto, iremos trabalhar com o WordPress.com mesmo sem a finalidade de lançar o *site* para a *internet*, ou seja, sem um servidor. Iremos trabalhar com um servidor local *offline*. O objetivo é se apoderar e tomar uso de todas as possibilidades da plataforma.

Após a instalação do WordPress.com no computador, a primeira coisa a fazer é escolher um tema. A própria plataforma se inicia com um tema padrão para desenvolvimento, mas se o usuário quiser, ele pode fazer a troca em qualquer momento do desenvolvimento de sua página.

Uma das vantagens do CMS são seus *plug-ins*<sup>24</sup>. Um *plug-in* é um software que contém um grupo de funções que podem ser adicionadas a um *site* WordPress. Eles podem estender a funcionalidade ou adicionar novos recursos aos seus *sites*. Eles são escritos na linguagem de programação PHP e se integram perfeitamente com a plataforma. Eles tornam mais fácil para os usuários adicionar recursos ao seu *site* sem conhecer uma única linha de código. Existem milhares destes disponíveis gratuitamente para *download* no diretório oficial do *plug-in* do WordPress.

Na comunidade WordPress, há um ditado que diz: "há um *plug-in* para isso". Existe *plug-in* para dinamizar a velocidade do *site* (W3 Total Cache<sup>25</sup>), te proteger se *spams* em comentários (AKISMET<sup>26</sup>), para compartilhamento em redes sociais (Flare<sup>27</sup>), organizar arquivos (Smart Archives Reloaded<sup>28</sup>) e até para montar a área de contato com um formulário editável (Contact Form 7<sup>29</sup>).

### 7.1.1. INSTALANDO OS *PLUG-INS*

Nesta seção iremos ensinar como instalar *plug-ins* em *site* WordPress. A forma de instalação se aplica a qualquer *plug-in*. No canto esquerdo da página inicial de desenvolvimento chamada "painel", haverá uma opção escrita "Plugins". Nessa página, o

---

<sup>24</sup> Disponível em: <<https://wordpress.org/plugins/>>. Acesso em: 20 nov. 2017.

<sup>25</sup> Disponível em: <<https://wordpress.org/plugins/w3-total-cache/>>. Acesso em: 19 nov. 2017.

<sup>26</sup> Disponível em: <<https://wordpress.org/plugins/akismet/>>. Acesso em: 19 nov. 2017.

<sup>27</sup> Disponível em: <<https://wordpress.org/plugins/flare/>>. Acesso em: 19 nov. 2017.

<sup>28</sup> Disponível em: <<https://wordpress.org/plugins/smart-archives-reloaded/>>. Acesso em: 19 nov. 2017.

<sup>29</sup> Disponível em: <<https://wordpress.org/plugins/contact-form-7/>>. Acesso em: 19 nov. 2017.

usuário terá acesso para gerenciar todos os *plug-ins* instalados. Na parte superior, haverá um botão com nome “Adicionar Novo”. Após clicá-lo, no canto direito haverá o campo de busca. Para considerar as *features* de acessibilidade para *web* mencionadas no capítulo 5. Sites correlatos, como aumentar e diminuir fontes, contraste de cores de fundo, pular para o conteúdo principal, destacar e sublinhar *links*, janela com intérprete virtual de LIBRAS e multilinguismo, escolhemos trabalhar com três *plug-ins*: *Google Language Translator*, *Hand Talk - WordPress* e *WP Accessibility Helper*.

#### **7.1.1.1. GOOGLE LANGUAGE TRANSLATOR<sup>30</sup>**

Este *plug-in* gratuito da Google funciona exatamente como sua ferramenta de tradução. Automaticamente ele traduz todo o conteúdo para qualquer idioma que o usuário selecionar. Para instalar, busque-o pela opção “Plugin” na barra lateral. Clique no canto inferior direito. Veja na Figura 17 abaixo:

---

<sup>30</sup> Disponível em: <<https://wordpress.org/plugins/google-language-translator/>>. Acesso em: 22 ago. 2017.

FIGURA 17 – Página do plug-in *Google Language Translator*



Google Language Translator

Descrição Instalação FAQ Registro de alterações Telas Comentários

Settings include: inline or vertical layout, show/hide specific languages, hide/show Google toolbar, and hide/show Google branding. Add the shortcode to pages, posts, and widgets.

**Versão:** 5.0.33  
**Autor:** [Rob Myrick](#)  
**Última atualização:** 1 mês atrás  
**Necessita a versão do WordPress:** 2.9 ou superior  
**Compatível até:** 4.8.2  
**Instalações ativas:** 100.000+  
[Página do plugin no WordPress.org »](#)  
[Página do plugin »](#)

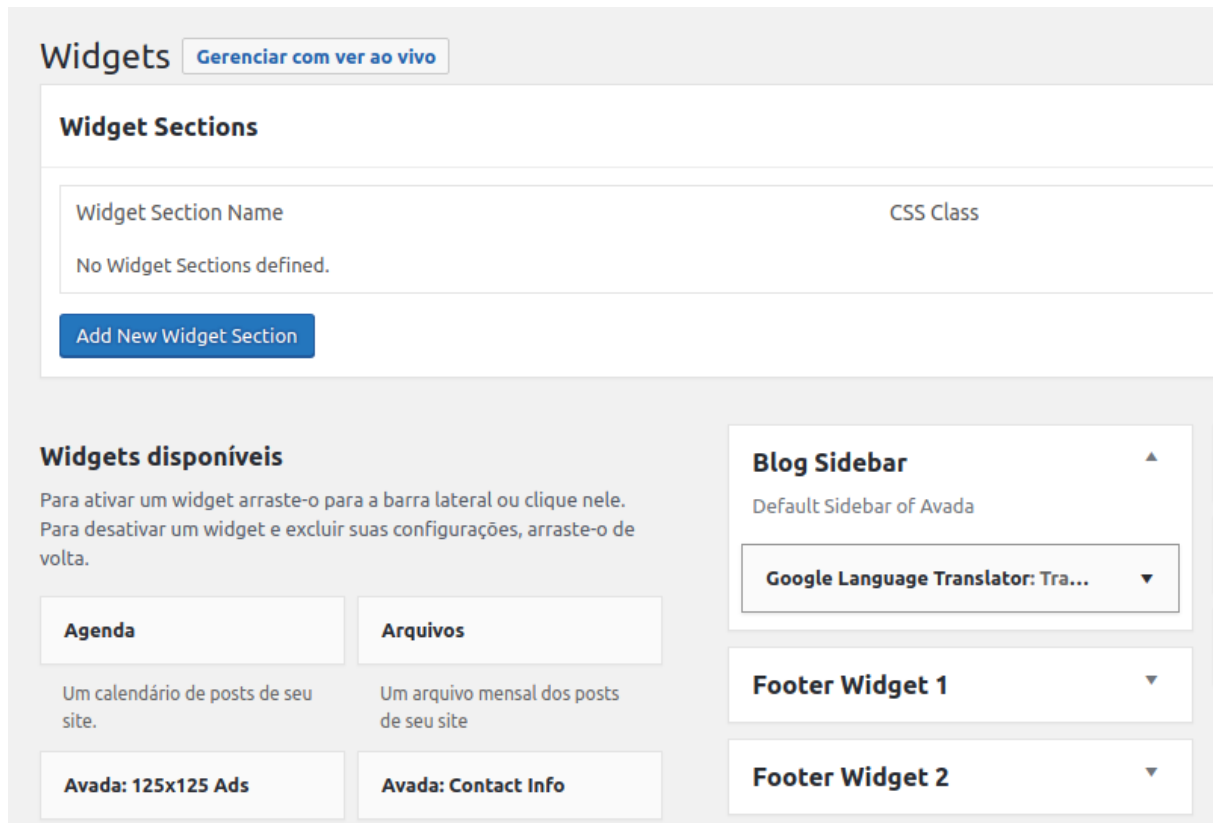
Instalada a versão mais recente

Fonte: Elaborada pelo autor (captura de tela).

Após a instalação, é preciso configurar. Na janela de configurações, é preciso ativar o *plug-in*, escolher o idioma original do *site*, especificar ou não para quais idiomas a página será traduzida e escolher opção de mostrar ou não as bandeiras. É recomendável que permita que o resto fique na situação padrão já sugerida pelo *plug-in*, mas tudo pode ser customizado.

Para inserir o botão de escolha de idioma na página, é preciso acessar o “Widgets” em “Aparência” na barra lateral, arrastar o *widget* Google Language Translator para alguma área de navegação como “Barra Lateral” ou “Rodapé”.

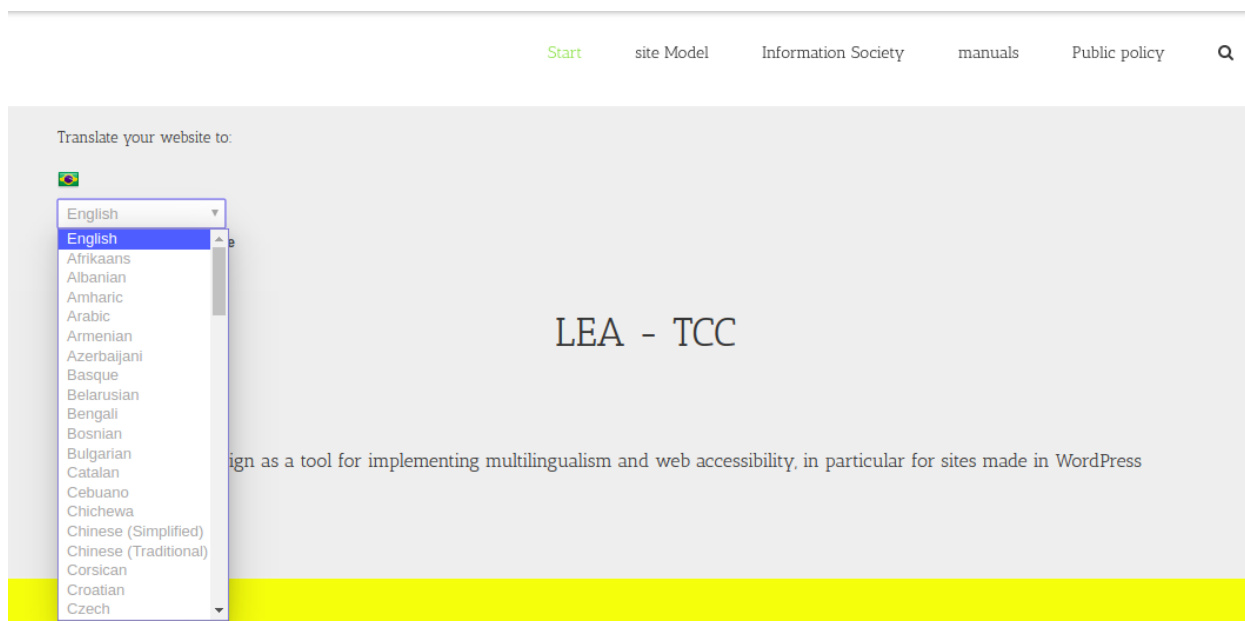
**FIGURA 18** – Página de widgets do WordPress



Fonte: Elaborada pelo autor (captura de tela).

Apesar da automatização da tradução do *site* inteiro, a desvantagem de *plug-ins* é que se o criador do *site* mexer no código fonte do *plug-in* para, por exemplo, melhorar a precisão de tradução, a ferramenta não seguirá as atualizações de seu criador. Sistemas de Gerenciamento de Conteúdo são extremamente limitados no quesito liberdade de programação. É indiscutível a otimização da criação de *sites* por pessoas com nenhum conhecimento de programação, porém, para aqueles que já desbravam o caminho da codificação, o sistema pode não ser eficiente para o desenvolvimento que planeja. Veja na Figura 19 a baixa precisão de tradução do *plug-in* por nós escolhido.

**FIGURA 19** – Página inicial do site exemplo em *WordPress*



Fonte: Elaborado pelo autor.

#### **7.1.1.2. HAND TALK - WORDPRESS<sup>31</sup>**

*Hand Talk* é uma ferramenta desenvolvida em 2002 com a finalidade de realizar traduções digitais e automáticas da língua portuguesa para a Língua Brasileira de Sinais - LIBRAS, utilizada pela comunidade surda e ensurdecida. O sistema age também como um auxiliador para intérpretes de comunicação entre surdos e ouvintes. Segundo o *site* oficial<sup>32</sup>, “a empresa premiada internacionalmente e referência no segmento, é comandada por um simpático intérprete virtual, o Hugo, personagem 3D que torna a comunicação mais interativa e de fácil compreensão” (HAND TALK).

A ferramenta foi desenvolvida tanto como *plug-in* de WordPress como implementação de código para *sites* desenvolvidos por programação. Nas duas plataformas, ele é pago. Por consequência, não foi possível visualizar a ferramenta em uso no *site* exemplo, porém é tranquilamente possível visualizar a ferramenta em ação em outras páginas de *web*, CMS ou não.

Para fazer a instalação no WordPress, como com qualquer outro *plug-in*, busque-o pela opção “Plugin” na barra lateral no painel do *site*. Clique no canto inferior direito. Veja na imagem abaixo:

<sup>31</sup> Disponível em: <<https://wordpress.org/plugins/handtalk/>>. Acesso em: 22 ago. 2017.

<sup>32</sup> Disponível em: <<https://handtalk.me/>>. Acesso em: 20 nov. 2017.

**FIGURA 20** – Página do *plug-in Hand Talk* - WordPress

Plugin oficial para Wordpress

hand talk

Torne o seu site acessível em Libras, automaticamente!

assine em: [www.handtalk.me/sites](http://www.handtalk.me/sites)

Hand Talk

Descrição Instalação FAQ Registro de alterações Telas Comentários

**Atenção:** Esse plugin não foi testado com sua versão atual do Wordpress.

Tornamos o mundo mais acessível com a ajuda do Hugo, nosso intérprete virtual. Conte com a maior plataforma de tradução automática do mundo em Língua de Sinais.

O Tradutor de Sites é representado por um botão de acessibilidade que fica do lado direito da tela. Quando ativado, o Hugo traduz os textos selecionados pelos visitantes para Libras, automaticamente. Esses sites acessíveis passam a abrir portas para milhões de pessoas, que até então

**Versão:** 2.0  
**Autor:** [Hand Talk](#)  
**Última atualização:** 1 ano atrás  
**Necessita a versão do Wordpress:** 3.7 ou superior  
**Compatível até:** 4.5.10  
**Instalações ativas:** 100+  
[Página do plugin no WordPress.org »](#)  
[Página do plugin »](#)

Instalada a versão mais recente

Fonte: Elaborada pelo autor (captura de tela).

Usando como amostra a aplicação no *site* exemplo, destaca-se aqui uma outra desvantagem dos *plug-ins* de CMS. Quanto maior for a utilização dessas ferramentas, maior será sua manutenção. A mensagem “Atenção: Esse plugin não foi testado com sua versão atual do Wordpress” indica que conforme a plataforma atualiza, os desenvolvedores dos *plug-ins* também precisam atualizar seu conteúdo e quem desenvolve os *sites* precisa atentar-se às atualizações para que todo seu funcionamento não seja corrompido. Em outras palavras, não é possível que seu *site* seja autossuficiente. É imprescindível o acompanhamento das atualizações de seus componentes para que tudo continue funcionando.

Para configurar o Hand Talk, clique em “Configurações” na barra lateral da página Painel e apenas insira o seu *token* adquirido após fazer o cadastramento no *site* oficial da ferramenta na caixa de propriedades como mostra a imagem abaixo:



**FIGURA 21** – Caixa de propriedades do *plug-in Hand Talk - WordPress*



**Hand Talk**

**Hand Talk Configuration**

To subscribe and get your token, go to [Hand Talk](#) website

**Propriedades**

```
{  
  "token": "INSIRA SEU TOKEN, AQUI!"  
}
```

*Formato JSON*

Para maiores informações, acesse nossa [Documentação](#)

Fonte: Elaborada pelo autor (captura de tela).

Após inserir o *token*, haverá um botão azul com duas mãos do lado direito da tela. Ao clicá-lo, a janela com o Hugo, o intérprete virtual 3D, aparecerá. O cursor do mouse mudará para duas mãos juntas quando passá-lo em cima de algum texto. Isso significa que quando clicar, o Hugo começará a fazer a tradução. Veja imagem abaixo.

**FIGURA 22** – Intérprete virtual do *plug-in Hand Talk - WordPress*



Fonte: Elaborada pelo autor (captura de tela).

Apesar da automatização e a contribuição para a comunidade surda e ensurdecida brasileira, a ferramenta ainda é limitada. Assim como qualquer outra língua estrangeira, LIBRAS tem suas variações, porém a ferramenta não contempla todas elas, além disso, alguns sinais ainda não são cadastrados no banco de dados, logo, o intérprete precisa soletrar as palavras, o que pode causar algum desconforto para o entendimento do usuário com necessidade especial auditiva, pois ele pode ter nascido assim e nunca ter tido contato com o alfabeto da língua portuguesa. Soletrar para alguns surdos na intenção de fazê-los entender a língua é o mesmo que soletrar português para um nativo chinês.

#### **7.1.1.3. WP ACCESSIBILITY HELPER<sup>33</sup>**

O *WP Accessibility Helper* é um *plug-in* interessante para as boas práticas de acessibilidade na *web*, além de ser gratuito e de fácil configuração. Para sua instalação, busque-o na aba “Plugins” na barra lateral do WordPress. Clique no canto inferior direito.

---

<sup>33</sup> Disponível em: <<https://wordpress.org/plugins/wp-accessibility-helper/>>. Acesso: 22 ago. 2017.

**FIGURA 23** – Página do *plug-in WP Accessibility Helper*

WP Accessibility Helper (WAH)

Descrição Instalação FAQ Registro de alterações Telas Comentários

WP Accessibility Helper sidebar provides the basic accessibility tools and helps your site to be much more accessible.

**Official website**

<https://accessibility-helper.co.il/>

**Upgrade notice:**

If you upgrade from 0.5.6.0 version (or lower) you need delete plugin and reinstall it again.

**Versão:** 0.5.8.4

**Autor:** [Alexander Volkov](#)

**Última atualização:** 2 meses atrás

**Necessita a versão do WordPress:** 4.0.1 ou superior

**Compatível até:** 4.8.2

**Instalações ativas:** 3.000+

[Página do plugin no WordPress.org »](#)

[Página do plugin »](#)

Instalada a versão mais recente

Fonte: Elaborada pelo autor (captura de tela).

Sua grande utilidade se dá pelo fato de ser uma ferramenta que reúne várias *features* em prol da acessibilidade na *web* como aumento e diminuição de fonte, escolha da cor de fundo como contraste, sublinhar e destacar *links*, inverter cores, alterar a escala de cinza e até remover animações.

Para sua configuração, clique na opção “Accessibility” que aparecerá na barra lateral de navegação do CMS. Todas as possibilidades de *features* estarão disponíveis para serem ou não selecionadas. Para este trabalho, customizamos apenas as *features* previamente mencionadas no capítulo 5. Sites correlatos. Veja na Figura 24:

**FIGURA 24** – Página de configuração do *plug-in WP Accessibility Helper*

The image shows the settings page for the WP Accessibility Helper plugin, organized into three main sections: Font Settings, Contrast Settings, and Styles Settings. Each section has a dark blue header with a downward arrow icon.

- Font Settings:**
  - Enable Readable Font?**: A toggle switch currently set to OFF.
  - Choose custom font**: A dropdown menu showing "Arial, Helvetica, sans-serif".
  - Choose font resize option**: A dropdown menu showing "Script base resize".
  - Font resize - title**: An empty text input field.
  - Reset font size - title**: An empty text input field.
  - \*\* This field work only when script base resize option chosen in "Choose font resize option"
- Contrast Settings:**
  - Enable contrast mode?**: A toggle switch currently set to ON.
  - Contrast - title**: A text input field containing "Contraste de Cor".
  - Choose color button - title**: A text input field containing "Escolha Cor".
  - Contrast variations?**: A toggle switch currently set to DEFAULT.
- Styles Settings:**
  - Remove animations mode?**: A toggle switch currently set to OFF.

Fonte: Elaborada pelo autor (captura de tela).

Na página do *site*, haverá um ícone quadrado com a silhueta de um cadeirante no centro no canto superior direito da tela. Por padrão, o botão é reconhecido por um programa de leitor de tela, caso o usuário precise. Ele é intuitivo e há a opção de customizar a tradução do título das opções que irão aparecer quando clicar no botão. No *site* exemplo deste projeto, posicionamos o botão ao outro lado da tela para que as opções, quando deslizadas para baixo, não sobrepussem a janela do intérprete virtual de LIBRAS.

**FIGURA 25** – Resultado final do site construído em *WordPress*



Fonte: Elaborada pelo autor (captura de tela).

### 7.1.2. CONCLUSÕES DO CMS

Apesar de *WP Accessibility Helper* ser um *plug-in* muito autossuficiente para as práticas de acessibilidade *web*, não foi possível criar a *feature* de pular para conteúdo principal com ele. Entretanto, com alguns temas do *WordPress* que promove a possibilidade de desenvolver um *site* estilo *one-page* é possível criar a função da *feature* por meio dos itens do menu. Quando o usuário clica em algum item da barra de navegação global, a página irá automaticamente até a posição onde se encontra o conteúdo, simulando a mesma função que a *feature*.

Após a conclusão da construção das ferramentas para um *site* desenvolvido com um CMS, veja no quadro abaixo como ficou o resultado da capacidade de acessibilidade do *WordPress*:

**QUADRO 1** – Resultado da capacidade de acessibilidade do *WordPress*

	WordPress
Aumento e diminuição de fonte	SIM
Contraste de cores de fundo	SIM
Pular para o conteúdo	NÃO
Destacar e sublinhar <i>links</i>	SIM
Janela com intérprete virtual de LIBRAS	SIM
Multilinguismo	SIM

Fonte: Elaborado pelo autor.

## 8. FRAMEWORK

Em geral, um *framework* é uma estrutura real ou conceitual destinada a servir de suporte ou guia para a construção de algo que expande a estrutura em algo útil. Um *framework* pode ser um conjunto de funções dentro de um sistema e como elas se inter-relacionam; as camadas de um sistema operacional; as camadas de um subsistema de aplicação; como a comunicação deve ser padronizada em algum nível de uma rede e assim por diante.

Marc Clinton (2003), criador de dois projetos de código aberto, o MyXaml, um mecanismo de instanciação declarativa e a estrutura de Testes de Unidade Avançada, e a Interacx, uma suíte de aplicativos multicamada comercial, explica, no *site Code Project For Those Who Code*<sup>34</sup>, que um *framework* facilita um trabalho com tecnologias complexas. Ele une vários componentes em algo mais útil, requer que o desenvolvedor implemente o código de forma a promover codificação consistente, promove menos *bugs* e aplicativos mais flexíveis, simplifica os testes e propõe uma depuração mais eficaz.

Clinton (2003) especifica a ferramenta em três grandes conjuntos de classificações, tais como:

- i. pacote que simplifica uma interface para uma tecnologia, reduz e elimina tarefas repetitivas, aumenta a flexibilidade das aplicações por meio da abstração e são muito reutilizáveis, independentemente das considerações de *design* de interface para uma tecnologia.
- ii. arquitetura que gerencia uma coleção de objetos discretos e implementa um conjunto de *design* específico.
- iii. metodologia que reforça adesão a uma abordagem de projeto consistente, desacopla dependências de objeto e são muitas vezes reutilizáveis independentemente dos requisitos da aplicação.

O desenvolvimento do site usando um *framework* é um processo intermediário no que diz respeito aos níveis de dificuldade descritos mais cedo. Certas linhas ou até mesmo blocos de códigos já são previamente prontos, o que dinamiza o processo de desenvolvimento. O *framework* a ser trabalhado nesse trabalho de conclusão de curso será o Bootstrap.

---

<sup>34</sup> Disponível em: <<https://www.codeproject.com/Articles/5381/What-Is-A-Framework>>. Acesso em: 07 out. 2017.

## 8.1. BOOTSTRAP

O Bootstrap é um *framework* de *front-end* para *web* livre e de código aberto para projetar *sites* e aplicativos na *web*. Ele contém modelos de *design* baseados em HTML e CSS para tipografia, formas, botões, navegações e outro componentes de interface, bem como extensões de JavaScript (principal linguagem de programação para o desenvolvimento do *layout* das páginas *web* que facilita a interação do cliente). Ao contrário de muitos *frameworks* para *web*, ele é focado apenas no desenvolvimento de *front-end*. Estima-se que mais de 7 milhões de *sites* usam a ferramenta. De acordo com a Wikipedia (2017), o Bootstrap é o segundo projeto bem mais destacado no GitHub (plataforma de hospedagem de código de projetos privados ou de código aberto utilizado por programadores do mundo inteiro)<sup>35</sup> com 111.600 estrelas e 51.500 *forks*. Ele é rápido para configurar, é fornecido em uma abundante documentação e fáceis instruções sobre como usar sua estrutura. Sua equipe está continuamente lançando novas versões e atualizações e promoveu novos recursos focados em acessibilidade na versão 3.x. Durante o desenvolvimento deste trabalho, Bootstrap está em sua versão 4.

### 8.1.1. INSTALANDO O BOOTSTRAP

Neste capítulo, iremos estudar a forma de instalar o *framework* e como usá-lo a fim de atender a lista de ferramentas criadas no quinto capítulo deste trabalho. Inicialmente se faz necessária a instalação do *framework*. Segundo a documentação<sup>36</sup> da ferramenta em questão, existem várias formas de efetuar sua instalação. É possível:

- i. fazer o download dos arquivos compilados em CSS e JS para mais tarde adicioná-los a seu projeto;
- ii. fazer o download dos arquivos fontes criando seu próprio *assets pipeline* (um pipeline de ativos fornece uma estrutura para concatenar e minimizar ou compactar ativos de JavaScript e CSS) e baixando os arquivos SASS (linguagem de estilo compilada ou interpretada por CSS), JavaScript e sua documentação de origem;
- iii. por meio de gerenciadores de pacote como npm, RubyGems, Composer, Bower e NuGet;
- iv. ou utilizar CDN que será a finalidade deste trabalho.

CDN, que em inglês significa *Content Delivery Network* (Rede de Fornecimento de Conteúdo, Rede de Distribuição de Conteúdo ou Rede de Entrega de

---

<sup>35</sup> Disponível em: <<https://github.com/twbs/bootstrap>>. Acesso em: 20 nov. 2017.

<sup>36</sup> Disponível em: <<https://getbootstrap.com/>>. Acesso em: 06 set. 2017.

Conteúdo), é um sistema que interliga computadores em uma rede por meio da *internet*. Um determinado conteúdo pode ser compartilhado de modo transparente a usuários finais. É um ótimo sistema para transferir grandes mídias sem comprometer suas performances (WIKIPEDIA, 2017).

Uma das vantagens de usarmos o sistema de rede mencionado a outros métodos de instalação é a simplicidade em adicionar algumas linhas de código na programação e obter uma ótima performance para alcançar o objetivo desejado. Salienta-se também o intuito principal deste trabalho na busca de formas mais simples para realização do site acessível. Logo, para um programador que ainda está em fase iniciante em seus estudos será mais descomplicado utilizar CDN a ter que configurar e dedicar-se mais tempo às outras formas de instalação.

Por convenção, logo ao final do código, próximo das tags `</body>` e `</html>` deve-se utilizar as tags `<link>` e `<script>` para conectar-se às bibliotecas e assim poder usá-las no projeto. Após a inserção do código, já é possível fazer uso dos atributos e classes do Bootstrap.

**FIGURA 26** – Código CDN do Bootstrap

```
285
286     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css"
287           integrity="sha384-Y6pD6FV/Vv2HJnA6t+vs1U6fwYXjCFTcEPhbNj0lyAFsXTsj8BfaDjzALEQsN6M"
288           crossorigin="anonymous">
289     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/js/bootstrap.min.js"
290           integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi047NrKGL03SEJAg45jXxnGIfYzk45i90RDIqNm1"
291           crossorigin="anonymous">
292     </script>
293 </body>
294 </html>
295
```

Fonte: Elaborada pelo autor (captura de tela).

É importante que ao longo do desenvolvimento de qualquer projeto haja um dos princípios mais importante da engenharia de *software*: planejamento. A acessibilidade precisa estar em desenvolvimento junto com todos os outros componentes do projeto. De fato, a acessibilidade deve ser um dos componentes do projeto. Deve ser um pensamento intuitivo: ao codificar um projeto, por trás dos panos a acessibilidade está inserida nesse pacote de planejamento. Em algum momento, todos os membros da equipe precisarão reunir-se e discutir as diretrizes para o projeto em questão.





### 8.1.2. PLUG-IN PAYPAL ACCESSIBILITY

Antes de darmos início ao desenvolvimento das *features* de acessibilidade, iremos trabalhar com um *plug-in* desenvolvido com e para o Bootstrap com a finalidade de beneficiar a navegação por teclado para as pessoas que possuem algum tipo de necessidade especial motora.

Para a realização do site acessível como produto-modelo deste trabalho, uma das formas utilizadas para a acessibilidade foi um *plug-in* criado pela *PayPal Accessibility*. Este *plug-in* foi desenvolvido com a finalidade de aprimorar e organizar a leitura de programas de leitor de tela para cegos, bem como sua navegação por teclado. Por meio de *mark-ups* adicionadas nos componentes do Bootstrap 3, leitores de tela terão uma melhor organização para lerem o conteúdo para usuários cegos. Os desenvolvedores do *plug-in* informam que a performance e o *layout* da página não serão prejudicados de forma alguma com a implementação dessa ferramenta. Seus criadores alegam também que os usuários de Bootstrap (versão 3.x) terão melhorias na navegabilidade e usabilidade sem dedicar esforço algum.

Para fazer sua instalação, ao entrar no repositório do GitHub, onde se encontra o código, faça o *download*<sup>37</sup> dos arquivos: “*bootstrap-accessibility.js*” e “*bootstrap-accessibility.min.js*”, como visto na Figura 27:

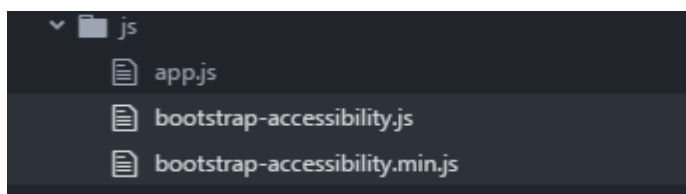
**FIGURA 27** – Arquivos de instalação do plug-in *Paypal Accessibility*

 <code>bootstrap-accessibility.js</code>	Fixes #104 - added jQueryUI license text in functions.js
 <code>bootstrap-accessibility.min.js</code>	Fixes #104 - added jQueryUI license text in functions.js

Fonte: Elaborada pelo autor (captura de tela).

Após o *download*, deve-se colocar os dois arquivos dentro da pasta de JavaScript para fazer a importação ao código do projeto visto nas imagens abaixo:

**FIGURA 28** – Arquivos do *plug-in Paypal Accessibility* na pasta JavaScript



Fonte: Elaborada pelo autor (captura de tela).

<sup>37</sup> Disponível em: <<https://github.com/paypal/bootstrap-accessibility-plugin>>. Acesso em: 01 set. 2017.

É preciso fazer a importação do *plug-in* ao código do projeto para que o seu funcionamento não seja comprometido:

**FIGURA 29** – Código CDN do *plug-in* Paypal Accessibility

```
279 <script src="http://code.jquery.com/jquery.js"></script>
280 <script src="/js/bootstrap.min.js"></script>
281 <script src="/js/bootstrap-accessibility.min.js"></script>
```

Fonte: Elaborada pelo autor (captura de tela).

Percebe-se na linha 279 da Figura 29 acima que há uma importação da biblioteca de jQuery para o funcionamento dos *plug-ins*. A codificação para o funcionamento do *plug-in* não será feita com jQuery, embora seja possível. O objetivo deste capítulo é exemplificar opções de acessibilidade apenas em Bootstrap. Os níveis de dificuldade visados neste capítulo devem ser mensurados pela capacidade de alcance de eficiência apenas do *framework* trabalhado. Iremos falar sobre a biblioteca jQuery mais à frente no momento que o nível de complexidade de desenvolvimento for necessário.

O *plug-in* foi criado para acessibilizar os seguintes componentes: caixa de alerta (*alert*), dica de contexto (*tooltip*), *popover*, modal com diálogo (*modal dialog*), menu suspenso (*dropdown*), painel guia (*tab panel*), caixa de informação expansível e galeria em carrossel (*carousel*). Como o foco deste trabalho não contempla daltonismo, não iremos trabalhar com caixa de alerta. Como não houve explicação suficiente na documentação do *popover* e *carousel*, não foi possível trabalhar com essas *features*. Iremos trabalhar, então, com *tooltip*, *modal dialog*, *dropdown*, *tab panel* e caixa de informação expansível.

#### 8.1.2.1. MODAL E TOOLTIP

Na área que abrange o *design* da interface do usuário para aplicativos computacionais, uma *modal* é um elemento de controle gráfico que se encontra dependente à janela principal de um aplicativo. Ao mesmo tempo que ele cria um modo que desabilita a janela principal – às vezes com fundo preto transparente –, ele se mantém visível com a janela modal como uma janela *child*<sup>38</sup> na frente de toda a aplicação. Antes que os usuários possam retornar ao aplicativo *parent*<sup>39</sup>, eles devem interagir com a janela modal. Isso evita

---

<sup>38</sup> Termo usado para designar a um elemento subordinado a um componente superior. No caso, o *parent*.

<sup>39</sup> Termo usado para designar a um elemento subordinante a um componente inferior. No caso, o *child*.

interromper o fluxo de trabalho na janela principal. Como muitas vezes os modais revelam uma caixa de diálogo, as janelas modais às vezes são chamadas diálogos modais.

A utilidade do *plug-in* da PayPal no modal se dá ao fato de que sem o primeiro, o leitor de tela ficará repetindo informação desnecessária para o usuário. Não haverá uma organização hierárquica de informações relevantes para o usuário o que comprometerá seu entendimento, além disso, no caso de um modal sem o *plug-in*, o leitor de tela não pausará a leitura após algum componente. Esse fator é de extrema importância para a navegabilidade do usuário para que ele tenha a possibilidade de ter um conhecimento prévio do conteúdo e decidir se ele é relevante para permitir o leitor de tela dar continuidade na leitura ou não.

Existem dois elementos para montar o modal e fazê-lo funcionar: o botão que atuará como um gatilho que fará o modal aparecer e o próprio modal. Um dos componentes previamente pronto do Bootstrap é o botão. É possível usar a tag de *link* `<a>` com o atributo *href*, atribuir *button* no atributo *type* e apenas depois nomear a *class* com um “btn” que um valor que pertence ao Bootstrap.

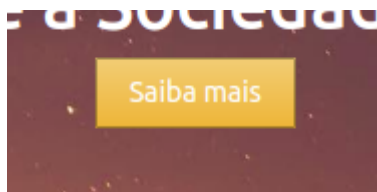
**FIGURA 30** – Código do botão do Bootstrap

```
83      <!-- Button trigger modal -->
84      <a href="#" type="button"
85          class="btn btn-warning"
86          data-toggle="modal"
87          data-target="#myModal">Saiba mais</a>
```

Fonte: Elaborada pelo autor (captura de tela).

No exemplo acima, além do nome “btn” para a classe, usamos “btn-warning” apenas para estética. O *layout* do botão será alaranjado fazendo referência a um botão de aviso.

**FIGURA 31** – Interface gráfico do botão do Bootstrap



Fonte: Elaborada pelo autor (captura de tela).

O atributo *data-toggle* pertence ao Bootstrap e é ele quem realizará o abrir e fechar do modal. O atributo *data-target* é quem faz referência ao *id* do modal. É ele quem dirá

quais dos possíveis modais no projeto que irá ser aberto no momento do clique do botão. É ele quem fará a conexão do botão com o modal.

Na imagem abaixo vemos um exemplo de um modal construído no *site-modelo* desenvolvido apenas em HTML e Bootstrap. Na linha 89, a classe “modal fade” é quem indica ao *framework* que o conteúdo abaixo aparecerá em uma janela modal que será chamada por um botão. O *id* é a identidade do modal. Ela deve ser única e não pode ser repetida para nenhum outro componente em todo o projeto, seja um modal ou não. Para o *plug-in* de acessibilidade começar a funcionar apropriadamente, certifique-se de adicionar “dialog” para o atributo *role*, “myModalLabel” para atributo *aria-labelledby* para referenciar o título do modal e por último, “true” ao atributo *aria-hidden* a fim de informar às tecnologias assistivas para ignorarem os elementos DOM do modal. Segundo o Wikipedia (2017), Modelo de Objeto de Documentos, em inglês *Document Object Model* (DOM), “é uma convenção multiplataforma e independente de linguagem para representação e interação com objetos em documentos HTML, XHTML e XML”.

**FIGURA 32** – Código do modal do Bootstrap

```
88 <!-- Modal -->
89 <div class="modal fade" id="myModal"
90     tabindex="-1"
91     role="dialog"
92     aria-labelledby="myModalLabel"
93     aria-hidden="true">
94   <div class="modal-dialog">
95     <div class="modal-content">
96       <div class="modal-header">
97         <h4 class="modal-title" id="myModalLabel">Esse é o título principal.</h4>
98       </div>
```

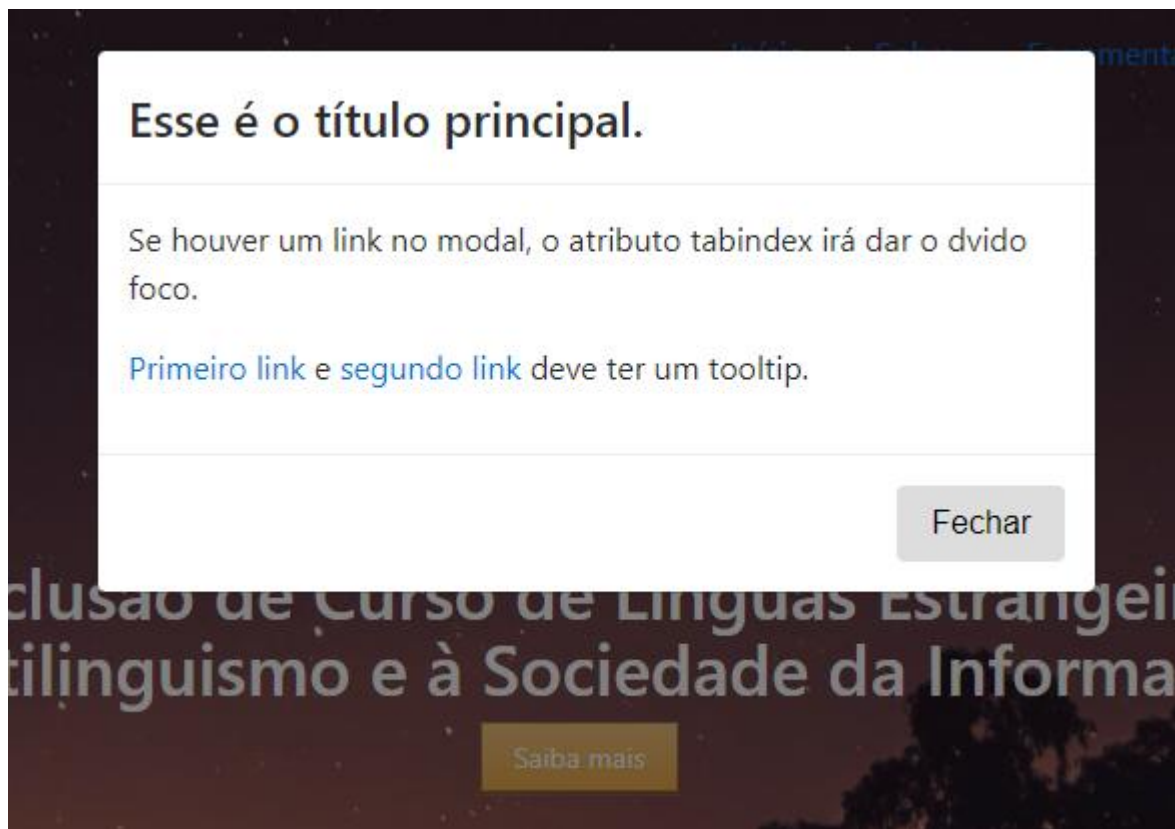
Fonte: Elaborada pelo autor (captura de tela).

As classes que se segue nas linhas seguintes são estipuladas por padrões do *framework*: “modal-dialog” para informar a existência de uma caixa de diálogo que irá interagir com o usuário; “modal-content” apenas para informar, dentro da hierarquia, o momento que se deve inserir o conteúdo do modal; “modal-header” para indicar onde será inserido o título do modal, se assim houver.

A classe “modal-title” e o *id* “myModalLabel” são para dar referência ao título para o leitor de tela. Ao abrir o modal, o leitor de tela irá iniciar lendo o título e informar se há um diálogo ou não para a interação do usuário. Na linha seguinte, a classe “modal-body” é

para informar o momento que deve inserir o corpo do modal com as tags <p> e <h1> se necessário. Veja na Figura 33 a interface gráfica do modal no site-modelo.

**FIGURA 33** – Interface gráfica do modal do Bootstrap



Fonte: Elaborada pelo autor (captura de tela).

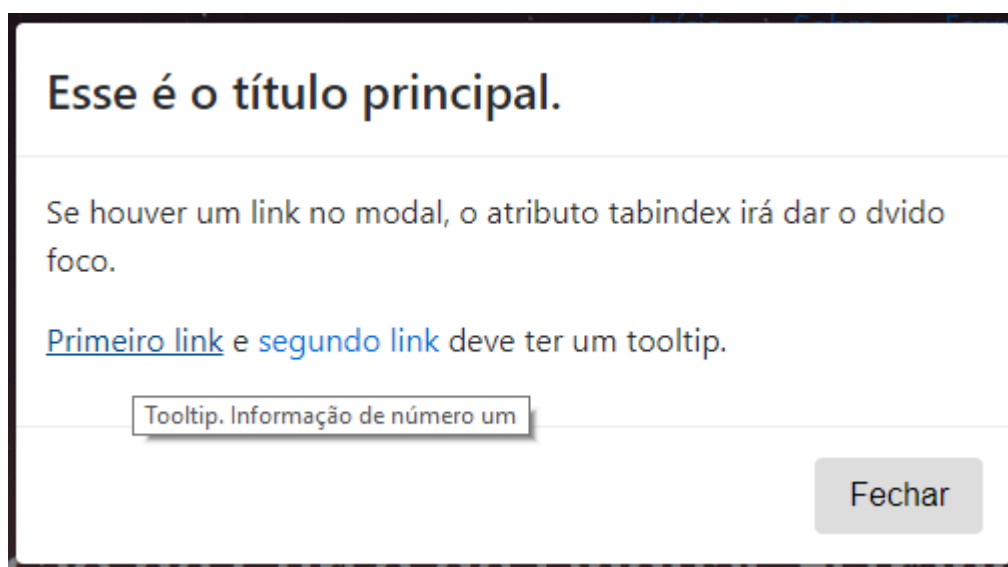
No exemplo deste site, existe uma amostra de um *tooltip* no corpo do modal. A dica de contexto, ou em inglês *Tooltip*, é um elemento muito comum de interface gráfica. Ele é uma pequena moldura flutuante que se abre quando o usuário passa o cursor do *mouse* em cima de um elemento de interface, normalmente uma palavra em um texto ou um *link*. A dica de contexto é muito utilizada em conjunto com o cursor. O usuário paira sobre um item sem clicar nele e uma pequena caixa flutuante aparece com informações complementares ao item apontado pelo cursor, uma informação ou explicação adicional sobre aquele elemento em específico. Abaixo veja o código do *tooltip* e sua interface gráfica.

**FIGURA 34** – Código do *tooltip* do Bootstrap

```
91 <div class="modal-body" tabindex="1" lang="pt">
92   <p>Se houver um link no modal, o atributo tabindex irá dar o dvido foco.</p>
93   <p><a href="#" class="tooltip-test" title="Tooltip. Informação de número um"
94     tabindex="1">Primeiro link</a>
95     e <a href="#" class="tooltip-test" title="Tooltip. Outra informação de número dois"
96       tabindex="1">segundo link</a> deve ter um tooltip.</p>
97 </div>
```

Fonte: Elaborada pelo autor (captura de tela).

**FIGURA 35** – Interface gráfica do *tooltip* do Bootstrap



Fonte: Elaborada pelo autor (captura de tela).

Com o *plug-in* ativado, é possível concluir que um leitor de tela conseguirá ler qualquer *tooltip* que estiver dentro do corpo do modal. Do contrário, o programa iria se confundir e acabar bagunçando toda a informação contida no modal. Os testes feitos no *site* desenvolvido apenas em Bootstrap nos levaram a concluir a importância do *plug-in*. Todo o conteúdo por trás do código é lido pelo NVDA – o leitor de tela –, o que não é necessariamente algo bom, uma vez que assim como um vidente tem a possibilidade de filtrar o conteúdo que lhe interessa possuindo um prévio acesso do conteúdo por meio da leitura, não é eficientemente informativo ser invadido com várias informações ao mesmo tempo por um leitor de tela, além das que se repetem por má interpretação do leitor. A pessoa cega deve ter essa mesma possibilidade de filtrar as informações e isso é possível se o leitor de tela ler um conteúdo e pausar para esperar o próximo comando do usuário. Assim, a pessoa terá o controle de continuar ouvindo o conteúdo lido pelo leitor ou dar sequência em sua navegação, por meio das teclas.

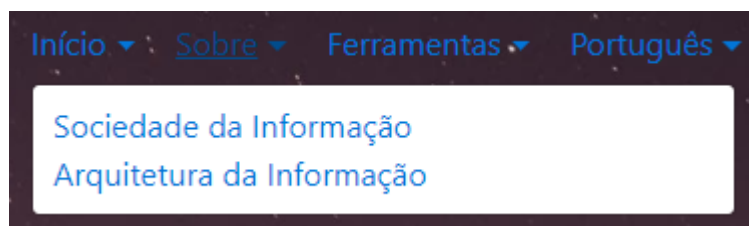
O *tooltip* é útil no momento em que o usuário pode lê-lo. A mensagem a aparecer na caixa flutuante será aquele valor atribuído a *title* nas linhas 93 e 95 da Figura 34. Por outro lado, para uma pessoa cega, a informação ainda será útil porque o leitor de tela irá ler aquele atributo. Pelos testes que fizemos, infelizmente é possível concluir que há uma falha no entendimento da informação do *tooltip* uma vez que o leitor de tela não avisa ao usuário que a informação a ser lida é um contexto de dica. Como *title* pode ser acessado bem como o *alt* para imagens, é importante usar-se das técnicas de audiodescrição e previamente informar do que se trata a informação a seguir. Por exemplo, no momento que o desenvolvedor estiver codificando o *tooltip*, é importante que digite “Tooltip.” ou “Dica de contexto” e então a informação. Assim, o leitor de tela irá ler e avisar ao usuário do que se trata a informação a seguir, se é uma dica de contexto ou uma imagem.

Após o conteúdo está a classe “modal-footer” para representar o rodapé do modal. Ele é composto por um elemento: um botão de fechar. Nesse elemento há um atributo chamado *data-dismiss* com nome de “modal”. Esse atributo é quem tem a função de fechar o modal, o empurrando para cima.

#### 8.1.2.2. DROPDOWN

Um *dropdown*, ou em português, lista suspensa ou lista de seleção, muito semelhante a uma caixa de listagem, é um elemento de controle gráfico que possibilita ao usuário escolher algum valor de uma lista. Quando um *dropdown* está inativo, ou seja, sem a lista aparente, ele exibe um único valor. Quando um usuário clica no único elemento aparente, uma caixa com várias opções posicionadas em lista vertical se exibe num movimento deslizante para baixo. Quando o usuário seleciona um novo valor, o controle retorna ao seu estado inativo, exibindo um único valor selecionável. Veja na imagem abaixo um exemplo de *dropdown* em uma barra de navegação.

**FIGURA 36** – Interface gráfica do *dropdown* do Bootstrap



Fonte: Elaborada pelo autor (captura de tela).

A utilidade do *plug-in* da PayPal no *dropdown* se dá ao fato de que sem o primeiro, o leitor de tela não poderá auxiliar o usuário cego a saber o que é sub menu e o que não é no momento que o *dropdown* se abrir para baixo. Já com o *plug-in*, o usuário não ficará tão perdido nos valores do *dropdown*.

A Figura 37 abaixo mostra o código do *dropdown* utilizado para o *site* modelo. Ele é uma referência de menu suspenso que se repete ao longo do projeto. Para evitar essa repetição de conteúdo, iremos trabalhar apenas com a estrutura deste menu com nome “Início” uma vez que ele possui a mesma estrutura que os outros menus.

**FIGURA 37** – Código do *dropdown* do Bootstrap

```
29      <ul id="links" href="#">
30        <div class="dropdown">
31          <li><a href="#" class="dropdown-toggle"
32                data-toggle="dropdown"
33                data-target=".inicio"
34                id="dropdownMenuInicio">Início</a></li>
35          <div class="inicio">
36            <ul class="dropdown-menu"
37                  role="menu"
38                  aria-labelledby="dropdownMenuInicio"
39                  aria-haspopup="true"
40                  aria-expanded="false">
41              <li><a href="#">Introdução</a></li>
42              <li><a href="#">Objetivo</a></li>
43              <li><a href="#">Justificativa</a></li>
44            </ul>
45          </div>
```

Fonte: Elaborada pelo autor (captura de tela).

Por padrão, o *framework* já fornece uma barra de navegação pronta, porém, para fins de esclarecimento teórico, decidimos construir uma *navbar* personalizada para ter mais liberdade na programação.

Usando a tag `<ul>`, criamos uma lista de elementos presente na barra de navegação, o menu suspenso “Início” é um deles. Na linha 30, criamos uma classe com nome “dropdown”. É ela quem indica que os componentes e elementos que seguem nas próximas linhas são os conteúdos do menu suspenso.

Na linha 31, temos a tag `<li>` que faz referência a lista iniciada com a tag `<ul>` na linha 29. Utilizamos as duas tags para criar a lista dentro da barra de navegação. A classe com nome “dropdown-toggle” e o atributo *data-toggle* são encarregados de efetuar a função de abrir e fechar a caixa do menu suspenso. O *data-target* é a referência para o botão abrir o determinado *dropdown* a escolha do programador. No exemplo do *site*, o atributo em questão



está referenciando a classe “.inicio”. Ou seja, ao clicar no botão que atuará como gatilho, o *dropdown* que será aberto será aquele que possuir uma classe com nome “inicio”.

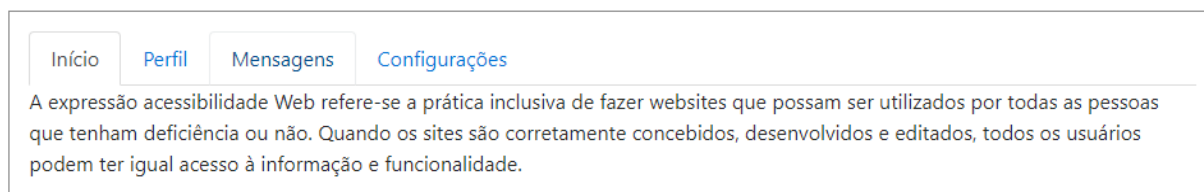
A classe “dropdown-menu” inicia o conteúdo dos valores do menu. Os atributos *aria-labelledby*, *aria-hidden* e principalmente o *aria-haspopup* são interpretados pelo *plug-in* e auxilia o NVDA – o leitor de tela – para dizer “sub-menu” para usuários cegos saberem que se trata dos valores de um menu suspenso pertencente a um elemento decretado como *dropdown*.

### 8.1.2.3. TAB

Uma interface de documento com abas (*Tabbed Document Interface* - TDI) ou Tab é um elemento de controle gráfico que automatiza em uma única janela vários documentos ou painéis. São muito parecidas com as janelas de navegação de *browsers*. Navegadores e aplicativos da *web* bem como editores de textos e painéis são comumente mais propícios a possuírem tabs.

Na Figura 38 abaixo, temos um exemplo da ferramenta aplicada no site modelo desenvolvido com Bootstrap:

**FIGURA 38** – Interface gráfica do *tab* do Bootstrap



Fonte: Elaborada pelo autor (captura de tela).

Para criar uma navegação como essa, é preciso primeiramente criar uma classe de nome “nav nav-tabs” para uma tag <ul> do HTML. Para que o *plug-in* funcione corretamente, é preciso incluir o atributo *role* com nome “tablist”. Ele informa que o elemento seguinte possui abas. Usando a tag <li> para o número de elementos que irão compor as abas, crie uma classe chamada “nav-item”. Insira a tag <a> com um *id* e atribua *href* para cada <li>. Nomeie o *href* com a *id* de cada respectiva tab. Em seguida, atribua o valor “tabi” a *role*. Insira *aria-controls* com o nome do *id* da *tab-pane*. Defina *aria-selected* como true, para que o conteúdo dessa aba seja o primeiro a ser reconhecido como selecionado pelo *plug-in*. Crie a classe padrão do *plug-in* “nav-link” e por último, o *data-toggle* como “tab” para fornecer a função de sumir e aparecer o conteúdo da aba.

**FIGURA 39** – Código do *tab* do Bootstrap

```
125      <!-- Nav tabs -->
126      <div class="bs-example">
127        <ul class="nav nav-tabs" role="tablist">
128          <li class="nav-item">
129            <a id="tab1" href="#home"
130              role="tab"
131              aria-controls="home"
132              aria-selected="true"
133              tabindex="0"
134              class="nav-link"
135              data-toggle="tab">Início</a></li>
```

Fonte: Elaborada pelo autor (captura de tela).

Após a criação das abas, é preciso criar a caixa que conterá os conteúdos referente a cada aba. Para isso, crie uma classe chamada “tab-content”, junto com *aria-live* com valor “polite”. É ele quem disponibilizará a navegação dentro das abas com a tecla tab do teclado.

Na linha seguinte, crie classes chamadas de “tab-pane”, “fade” e “active”. Elas se encarregaram de propor um efeito lento na transição de aparecer e sumir o conteúdo da caixa e forçar o texto da primeira aba iniciar selecionado, além do *layout* das tabs. Por último, criamos o *id* que será referenciado no atributo *href* da aba que foi criada anteriormente.

**FIGURA 40** – Código do conteúdo do *tab* do Bootstrap

```
161      <!-- Tab panes -->
162      <div class="tab-content" aria-live="polite">
163        <div class="tab-pane fade" id="home"
164          role="tabpanel"
165          aria-labelledby="tab1"
166          aria-hidden="false"
167          tabindex="0"> A expressão acessibilidade Web refere-se a prática inclusiva de fazer
168        websites que possam ser utilizados por todas as pessoas que tenham deficiência ou não.
169        Quando os sites são corretamente concebidos, desenvolvidos e editados, todos os usuários
170        podem ter igual acesso à informação e funcionalidade.
171      </div>
```

Fonte: Elaborada pelo autor (captura de tela).

Atribua o valor “tabpanel” para a *role* para que o *plug-in* possa reconhecer o conteúdo de aba. Adicione *aria-labelledby* com o *id* da tab e *aria-hidden* igual a “false” para que o conteúdo não inicie em exibição ativada.

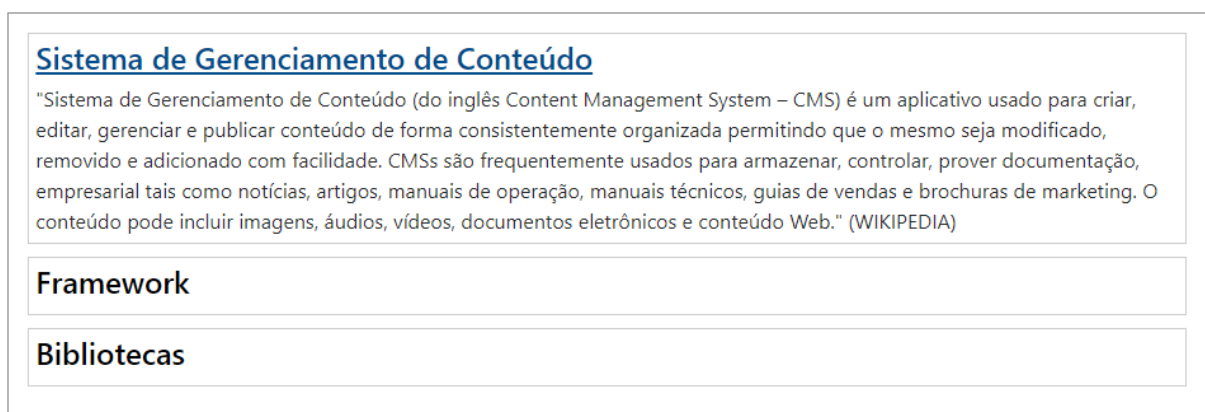
A utilidade do *plug-in* da PayPal no tab se dá ao fato de que sem o *plug-in*, não conseguiria navegar dentro das abas com a tecla tab do teclado. O usuário ainda

conseguiria navegar pelas abas, mas teria que se localizar com o cursor do *mouse* para cima do conteúdo para que o leitor de tela pudesse ler o conteúdo. Isso pode comprometer a efetividade da identificação da informação se o usuário não tiver experiência com o *mouse* ou se ainda for um usuário com baixa coordenação motora.

#### 8.1.2.4. COLLAPSE

O *collapse* ou *accordion* é um elemento de controle gráfico que compreende uma lista de itens empilhados verticalmente. Para exibir o conteúdo associado ao item desta lista, cada item pode ser "expandido" ou "esticado". Dependendo da configuração, pode ter nenhum ou vários itens para ser expandidos. Um *collapse* é semelhante a uma interface com abas, uma lista de itens onde um item é expandido para um painel. Veja um exemplo abaixo:

**FIGURA 41** – Interface gráfica do *collapse* do Bootstrap



Fonte: Elaborada pelo autor (captura de tela).

Para montar um *accordion* acessível a um leitor de tela é importante, primeiramente, entender que algumas classes irão mudar daquelas usadas pelo *framework* por padrão. Na Figura 42 abaixo pode-se ver que o código usado para construir um *collapse* escrito na documentação é diferente do código usado pelos desenvolvedores do *plug-in*. A diferença entre eles está exatamente na efetividade da acessibilidade.

**FIGURA 42** – Código do *collapse* do Bootstrap sem o *plug-in Paypal Accessibility*

```
<div id="accordion" role="tablist" aria-multiselectable="true">
  <div class="card">
    <div class="card-header" role="tab" id="headingOne">
      <h5 class="mb-0">
        <a data-toggle="collapse" data-parent="#accordion" href="#collapseOne" aria-expanded="true">
          Collapsible Group Item #1
        </a>
      </h5>
    </div>

    <div id="collapseOne" class="collapse show" role="tabpanel" aria-labelledby="headingOne">
      <div class="card-block">
        Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad s
      </div>
    </div>
  </div>
</div>
```

Fonte: Elaborada pelo autor (captura de tela).

A utilidade do *plug-in* da PayPal no tab se dá ao fato de que enquanto o código por *default* não consegue fazer com que o NVDA leia o conteúdo dentro da caixa, o desenvolvido pela *PayPal* consegue por meio da tecla de seta para baixo, indicando que ao abrir o painel referente àquele título, o leitor pode iniciar a leitura. Não o suficiente, em algum momento o leitor irá parar a leitura para que o usuário tenha a disponibilidade de dar a continuidade, ou não, caso esta não lhe interesse ou não remeta ao que ele procura.

Para criar um *collapse* acessível é preciso iniciar com uma classe chamada “collapse-example”. Na linha seguinte, uma classe chamada “panel-group” com um *id* igual a “accordion”. Em baixo, classes chamadas “panel panel-default”. Depois, outra classe chamada “panel-heading”. Após iniciar essas três classes que foram desenvolvidas por padrão pela *PayPal*, use a tag <h4> do HTML para iniciar o título da caixa clicável que irá abrir o painel de conteúdo deslizando para baixo. Nesta tag, crie mais uma classe padrão do *plug-in* chamada “panel-title”. Nomeie as classes deste *link* com “accordion-toggle”. É ela quem irá proporcionar a função de abrir e fechar o painel de conteúdo. Na mesma tag, use os atributos *data-toggle*, *data-parent* e *href* com os componentes “collapse”, “#accordion” e “#collapseOne” respectivamente. Estes atributos auxiliarão o manuseio do conteúdo por meio das teclas de navegação do *plug-in*. O último atributo fará referência ao *id* do painel de conteúdo.

**FIGURA 43** – Código do *collapse* do Bootstrap com o *plug-in Paypal Accessibility*

```
202 <!-- Collapse -->
203 <div class="collapse-example">
204   <div class="panel-group" id="accordion">
205     <div class="panel panel-default">
206       <div class="panel-heading">
207         <h4 class="panel-title">
208           <a class="accordion-toggle"
209             data-toggle="collapse"
210             data-parent="#accordion"
211             href="#collapseOne">
212             Sistema de Gerenciamento de Conteúdo
213           </a>
214         </h4>
215       </div>
```

Fonte: Elaborada pelo autor (captura de tela).

Para criar o painel que será aberto com o deslize para baixo, crie um *id* com nome “collapseOne” e classes chamadas “panel-collapse collapse in”. Crie uma classe chamada “panel-body” para iniciar o conteúdo. Para criar várias abas na vertical, repita o processo desde a classe “panel-group” na linha 204. Veja a Figura 44 abaixo:

**FIGURA 44** – Código do conteúdo do *collapse* do Bootstrap

```
216 <div id="collapseOne" class="panel-collapse collapse in">
217   <div class="panel-body">
218     "Sistema de Gerenciamento de Conteúdo (do inglês Content Management System - CMS)
219     é um aplicativo usado para criar, editar, gerenciar e publicar conteúdo de forma
220     consistentemente organizada permitindo que o mesmo seja modificado, removido e
221     adicionado com facilidade. CMSs são frequentemente usados para armazenar,
222     controlar, prover documentação, empresarial tais como notícias, artigos,
223     manuais de operação, manuais técnicos, guias de vendas e brochuras de marketing.
224     O conteúdo pode incluir imagens, áudios, vídeos, documentos eletrônicos e conteúdo Web." (WIKIPEDIA)
225   </div>
226 </div>
```

Fonte: Elaborada pelo autor (captura de tela).

Neste subitem terminamos com o desenvolvimento do *plug-in* para Bootstrap. Nos subitens a seguir, iremos trabalhar, ainda com o Bootstrap, com as *features* da lista que criamos no capítulo cinco deste projeto.

### 8.1.3. MULTILINGUISMO COM BOOTSTRAP

A documentação do Bootstrap em seu *site* oficial não menciona qualquer possibilidade de transformar um sítio eletrônico multilingue. Por conta da carência de conteúdo, fizemos uma pesquisa de campo na *web*. O principal meio de pesquisa foi o *GitHub*, que segundo a Wikipédia:

GitHub é amplamente utilizado por programadores para divulgação de seus trabalhos ou para que outros programadores contribuam com o projeto, além de promover fácil comunicação através de recursos que relatam problemas ou mesclam repositórios remotos (*issues*, *pull request*). O GitHub é mundialmente usado e chega a ter mais de 3 milhões de usuários ativos mundialmente contribuindo em projetos comerciais ou pessoais. Hoje o GitHub abriga mais de 65 milhões de projetos, alguns deles que são conhecidos mundialmente (2017).

Mesmo com um repositório com tamanha informação, por meio da pesquisa é passível de conclusão que não foi encontrado nenhuma biblioteca i18n – abreviatura para “*internacionalization*” – desenvolvida pelo *framework*. Muitos desenvolvedores mencionaram possibilidades de multilinguismo em *sites* feitos com Bootstrap usando alguns *templates* prontos (sem necessidade de programação) desenvolvidos pelo *framework*. Esse tipo de *site* multilingue não se aplica ao foco deste trabalho. Trabalhar com conteúdo de sítio eletrônico parcialmente pronto é de caráter dos CMS mencionados anteriormente e não de *framework*. Se houvesse a explicação e análise de cada *template* multilingue desenvolvido com Bootstrap, o trabalho se tornaria extenso e disperso.

Apesar da ausência de alguma biblioteca i18n em Bootstrap, há uma forma de tornar o *site* multilingue. Primeiramente, é importante entendermos que mesmo uma biblioteca ou *plug-in* que automatize o multilinguismo na *web*, ainda não há uma ferramenta que substitua um tradutor por completo. Em qualquer procedimento de tradução automatizada é sempre importante haver a presença de um profissional da área por perto. Partindo deste dado, o processo de tradução de um *site* em Bootstrap não será totalmente automatizada. Para tal processo é crucial a presença de um tradutor.

Como vimos, para uma ferramenta com desenvolvimento mais livre em comparação aos CMS, o Bootstrap se torna ineficiente e limitado. Embora o método que encontramos para transformar um *site* multilingue dentro dos limites de um *framework* seja baseado majoritariamente em HTML, a barra de navegação usada pelo Bootstrap é importante para a realização da *feature*. Como Morville e Rosenfeld (2006) afirmaram em *Information Architecture for World Wide Web*, a barra de navegação global, por convenção, aparece no topo do *site*. Partindo deste conhecimento, inserimos o botão de escolha da língua estrangeira na navegação global.

**FIGURA 45** – Barra de navegação do site em Bootstrap

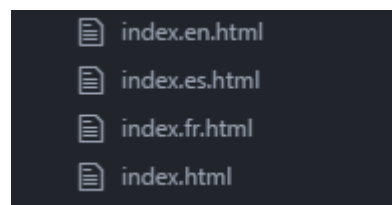


Fonte: Elaborada pelo autor (captura de tela).

A estrutura do botão de escolha de idioma foi feita com Bootstrap. Foi preciso criar um *dropdown* para que houvesse a possibilidade de várias opções. Cada opção corresponde uma língua estrangeira. Além do português – nossa língua nativa –, o recorte das línguas estrangeiras utilizadas no *site* foi com base àquelas ensinadas no nosso curso de formação, Línguas Estrangeiras Aplicadas ao Multilinguismo e à Sociedade da Informação. São elas inglês, francês e espanhol.

Foi necessária a construção da mesma página três vezes, cada uma com uma extensão diferente. Veja na Figura 46 abaixo:

**FIGURA 46** – Arquivos das páginas em inglês, francês e espanhol



Fonte: Elaborada pelo autor (captura de tela).

É preciso fazer a tradução manualmente por um profissional da área. Cada página pode ter o seu arquivo CSS e JavaScript. Após a criação e tradução do conteúdo para cada página, é preciso programar. Insira o atributo *lang* na tag `<html>` no início da página com a sigla do idioma de sua escolha. Assim, o leitor de tela saberá em qual língua está o *site*.

**FIGURA 47** – Código do atributo LANG para o multilinguismo com Bootstrap

```
1 <!DOCTYPE html>
2 <html lang="pt">
3 <head>
```

Fonte: Elaborada pelo autor (captura de tela).

Depois, é preciso criar um *dropdown* como na Figura 48 apresentada abaixo, onde no atributo *href* da tag `<a>` colocará o nome de cada página com sua devida extensão como um *link*.

**FIGURA 48** – Código do *dropdown* do Bootstrap

```
64      <li><a href="#" class="dropdown-toggle"
65              data-toggle="dropdown"
66              data-target=".languages"
67              title="é o idioma selecionado. Mudar de idioma."
68              id="dropdownMenuIdiomas">Português</a></li>
69      <div class="languages">
70          <ul class="dropdown-menu" role="menu" aria-labelledby="dLabel">
71              <li><a href="index.en.html" lang="en">English</a></li>
72              <li><a href="index.fr.html" lang="fr">Français</a></li>
73              <li><a href="index.es.html" lang="es">Español</a></li>
74          </ul>
75      </div>
```

Fonte: Elaborada pelo autor (captura de tela).

#### 8.1.4. PULAR PARA O CONTEÚDO

Na acessibilidade *web*, é crucial a dinamização de informação para o usuário. Morville e Rosenfeld (2006), no livro *Information Architecture for World Wide Web*, afirmam que a navegabilidade também se constrói com a capacidade de busca. Recorrer a um item específico que o usuário já sabe onde encontrar é uma capacidade para todos: videntes e cegos. Logo, é crucial que uma pessoa cega possa alcançar uma informação a qual ela já havia encontrado em outro momento com toda facilidade como uma pessoa vidente.

Para que a capacidade de busca contemple pessoas cegas, foi criado um botão que encaminhasse o usuário direto ao conteúdo do *site*. A posição deste botão é estratégica porque ele precisa estar antes da barra de navegação, ou pelo menos ser o primeiro item desta, porque assim será vantajoso para o usuário ouvir pelo leitor de tela e, caso decida, ser encaminhado direto para o conteúdo sem perder tempo navegando em uma área do *site* do qual ele já tem conhecimento que não fará uso ou que não seja pertinente para aquele momento.

É possível construir esse botão com o Bootstrap usando algumas tags *default* do HTML e outras da documentação do *framework*. É preciso apenas nomear o atributo *href* da tag *<a>* com o *id* da classe que inicia o conteúdo. No projeto, o botão foi inserido como primeiro item dentro da barra de navegação. O início do conteúdo contém um *id* com nome “main”. É por meio dessa referência em *href* que a funcionalidade irá direcionar o usuário direto ao conteúdo.



**FIGURA 49** – Código de pular para o conteúdo principal do Bootstrap

```
22      <!-- Navigation -->
23      <nav role="navigation" class="custom-navigation">
24          <a href="#main" id="logo" tabindex="2">Pular para o conteúdo</a>
```

Fonte: Elaborada pelo autor (captura de tela).

O *plug-in* da PayPal Accessibility é uma ferramenta muito benéfica para a organização da leitura de um leitor de tela. Sem o *plug-in*, muitas *features* mencionadas acabam confundindo o leitor de tela com sua organização. Algumas podem até ser acessadas pelo programa, mas não possuem uma boa organização para que o leitor tenha uma absorção de informação confortável. De fato, mesmo que ele não contemple diretamente nenhuma *features* da lista criada no capítulo 5, ela é de extrema relevância para uma navegação com o teclado, o que contribui indiretamente para o objetivo deste trabalho.

#### 8.1.5. CONCLUSÕES DO *FRAMEWORK*

Com os estudos sobre a documentação do *framework*, não foi possível encontrar formas de criar todas as ferramentas que contemplem a lista do capítulo 5. Sites correlatos. Bootstrap se limita a customizar os estilos e *layout* do *site* dinamicamente por meio de algum comando. É possível customizar seus componentes por meio do CSS programando o código previamente, mas não é possível fazer tal mudança com o disparo de algum botão durante o uso do *site*. Por esse empecilho, não conseguimos trabalhar com as *features* de acessibilidade, como destacar e sublinhar *links* bem como aumento e diminuição de fonte para usuários com baixa visão e janela virtual de LIBRAS para surdos e ensurdecidos. Veja abaixo o quadro com a possibilidade de acessibilidade com o *framework* Bootstrap:

**QUADRO 2** – Resultado da capacidade de acessibilidade do Bootstrap

	Bootstrap
Aumento e diminuição de fonte	Não
Contraste de Cores	Não
Pular para o conteúdo principal	Sim
Destacar e sublinhar <i>links</i>	Não
Janela com intérprete virtual de LIBRAS	Não

Multilinguismo	Sim
----------------	-----

Fonte: Elaborado pelo autor.

## 9. BIBLIOTECAS

Em ciência da computação, na maioria das vezes para o desenvolvimento de *software*, uma biblioteca é um acervo de recursos estáveis usados por programas de computador. Alguns exemplos destes recursos são dados de ajuda, modelos de mensagem, documentação, dados de configuração, código pré-escrito e sub-rotinas, valores, classes ou especificações de tipo. Segundo a Wikipédia:

[...] biblioteca é uma coleção de subprogramas utilizados no desenvolvimento de *software(sic)*. Bibliotecas contém código e dados auxiliares, que provém serviços a programas independentes, o que permite o compartilhamento e alteração de código e dados de forma modular (2016).

A maioria dos programas possuem alguma biblioteca que implementa os serviços do sistema e por ser um componente de desenvolvimento tão importante, decidimos buscar possíveis formas de automatizar as ferramentas de acessibilidade *web* por meio de bibliotecas.

A grande vantagem das bibliotecas é em questão de dinamização de código. Elas encurtam o código para que ele se torne mais limpo e de fácil sintaxe para compreensão. Elas são ferramentas extremamente poderosas que tornam qualquer site mais atrativo, melhorando sua navegabilidade e usabilidade. Qualquer site bem desenvolvido na *web*, hoje, possui alguma biblioteca. Logo, como em toda área de desenvolvimento de sistemas, existem bibliotecas tanto para o *front-end* como para o *back-end*. No site-exemplo deste trabalho, uma biblioteca muito popular entre os desenvolvedores foi usada para automatizar as *features* de acessibilidade: jQuery.

### 9.1. JQUERY

jQuery é uma rápida e pequena biblioteca rica em recursos de JavaScript. Isso torna os processos como manipulação de documentos HTML, manipulação de eventos e animação muito mais simples com uma interface fácil de usar que funciona em uma infinidade de navegadores. Com uma combinação de versatilidade e extensibilidade, jQuery mudou a forma como milhões de pessoas escrevem JavaScript.

Segundo o Craig Buckle, no SitePoint:

jQuery permanece sendo a biblioteca de JavaScript mais utilizada e já foi distribuída com o WordPress, o ASP.NET e vários outros frameworks. Ela revolucionou o desenvolvimento do lado do cliente, introduzindo o seletor CSS para a recuperação do nó DOM mais o encadeamento para aplicar manipuladores de eventos, animações e chamadas Ajax (2017, tradução nossa<sup>40</sup>).

A sintaxe do jQuery foi planejada para tornar a navegação em documentos mais fácil, dinamizar a criação de animações, manipular eventos bem como simplificar o *script* do lado do cliente. Para que os desenvolvedores criem abstrações para interação e animação de baixo nível assim como efeitos avançados como *widgets*, a jQuery possui recursos para criar *plug-ins* em cima da biblioteca de JavaScript. A abordagem modular da biblioteca jQuery permite a criação de poderosas páginas web dinâmicas e aplicativos da *web*.

Uma das grandes vantagens de jQuery é que ele é capaz de automatizar o CSS. Com a biblioteca, é possível customizar o conteúdo de CSS por meio de comandos do usuário, mesmo enquanto o *site* estiver em execução. Partindo desta informação por meio da biblioteca é possível customizar o *background-color*, *font-size* e *text-decoration* da página. Não foi preciso nenhum *plug-in* de JavaScript para codificar uma função que customizasse o CSS em modo de produção do *site*. Todas as funcionalidades foram executadas com poucas linhas de jQuery.

### 9.1.1. INSTALANDO JQUERY

Para inserir as funções jQuery é preciso fazer o *download* da ferramenta, porém, assim como Bootstrap, neste projeto iremos usar o CDN da biblioteca. Insere-se apenas o *link* da biblioteca com seus devidos atributos dentro de uma tag `<script>` de HTML.

FIGURA 50 – Código CDN do jQuery

```
372 <!-- jQuery first, then Tether, then Bootstrap JS. -->
373 <script src="https://code.jquery.com/jquery-3.1.1.slim.min.js"
374       integrity="sha384-A7FZj7v+d/sdmMqp/nOQwliLvUsJfDHW+k90mg/a/EheAdgtzNs3hpfag6Ed950n"
375       crossorigin="anonymous"></script>
```

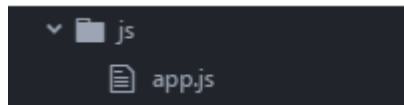
Fonte: Elaborada pelo autor (captura de tela).

Crie uma pasta com extensão *.js* dentro do seu projeto e faça a importação com a tag `<script>` de HTML. Veja nas imagens seguintes respectivamente:

---

<sup>40</sup> jQuery remains the most-used JavaScript library ever created and is distributed with WordPress, ASP.NET and several other frameworks. It revolutionized client-side development by introducing CSS selector to DOM node retrieval plus chaining to apply event handlers, animations, and Ajax calls.

**FIGURA 51** – Pasta com extensão .js para código jQuery



Fonte: Elaborada pelo autor (captura de tela).

**FIGURA 52** – Código de importação do arquivo app.js

```
378 <script src="js/app.js"></script>
```

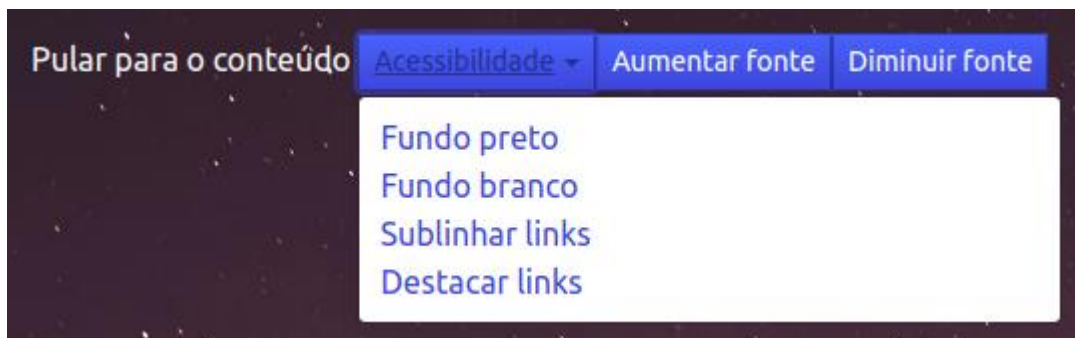
Fonte: Elaborada pelo autor (captura de tela).

O *id* dos elementos serão os gatilhos para a execução das funções do jQuery. Para isso, criamos um *id* para cada *feature* criada para os botões. Veja a lista abaixo:

- Fundo preto: change-bg-color-black;
- Fundo branco: change-bg-color-white;
- Sublinhar *links*: underline-links;
- Destacar *links*: highlight-links;
- Aumentar fonte: up-font-size;
- Diminuir fonte: down-font-size.

É importante salientar que a biblioteca em questão não é provida de nenhum pacote de customização predefinido para criação de botões, logo, foi preciso a mesclagem de ferramentas de jQuery com Bootstrap. Para criar a sequência de três botões no início do *site* deste trabalho, usamos a class “btn-group”. Para o primeiro botão “Acessibilidade”, usamos a classe “dropdown”. Os valores do *dropdown* foram “Fundo preto”, “Fundo branco”, “Sublinhar links” e “Destacar links”.

**FIGURA 53** – Interface gráfico do dropdown do Bootstrap para jQuery



Fonte: Elaborada pelo autor (captura de tela).

### 9.1.2. SUBLINHAR E DESTACAR LINKS

No arquivo de extensão .js que contará com todo o código JavaScript e jQuery. Por meio da biblioteca, selecione um *id*, que será o gatilho, com o método “.click”. Após o clique, o elemento *html* terá seus atributos CSS mudados com o método “.css”. Esse método utiliza dois argumentos: o atributo a ser mudado e a mudança. No caso do elemento *html*, terá seu “background-color” alterado para “black” quando o usuário clicar no botão “Fundo preto” com *id* “change-bg-color-black. O seletor do jQuery não trabalha apenas com tags de HTML. É possível alterar uma classe inteira criada pelo desenvolvedor. Com o mesmo *id*, é possível mudar o fundo do elemento *html* para preto e a classe “the-text” para uma cor branca.

Selecionando o elemento a ser alterado por meio do \$, aplique neste elemento o método .css com dois argumentos. Veja na Figura 54 que os *ids* “highlight-links” e “underline-links” são os gatilhos para alterar respectivamente a cor do *background* e a decoração do texto da classe “highlight”.

**FIGURA 54** – Código jQuery para botões de sublinhar e destacar links e contraste de cor de fundo

```
1 $(document).ready(function() {  
2  
3     $('#change-bg-color-black').click(function() {  
4         $('html').css('background-color', 'black');  
5         $('.the-text').css('color', 'white')  
6     });  
7  
8     $('#change-bg-color-white').click(function() {  
9         $('html').css('background-color', 'white');  
10        $('.the-text').css('color', 'black')  
11    });  
12  
13    $("#highlight-links").click(function() {  
14        $('.highlight').css('background-color', 'yellow');  
15    });  
16  
17    $("#underline-links").click(function() {  
18        $('.highlight').css('text-decoration', 'underline');  
19    });  
20 })
```

Fonte: Elaborada pelo autor (captura de tela).

### 9.1.3. BOTÃO PARA AUMENTAR E DIMINUIR FONTE

Para os botões de aumentar e diminuir a fonte da letra do *site*, foi preciso do *id* “up-font-size”, a tag <h1> do HTML e as classes “the-text” e “highlight”. O atributo “fontSize” receberá uma função que armazenará o valor padrão dentro de “parseFloat” e multiplicará por 1,15 para aumentar e dividirá por 1,15 para diminuir. Veja na Figura 54 e na Figura 55 o aumento e a diminuição respectivamente.

**FIGURA 55** – Código jQuery para botão de aumento de fonte

```
21 $( "#up-font-size" ).on( "click", function() {  
22     $('h1').css({  
23         fontSize: function( index, value ) {  
24             return parseFloat( value ) * 1.15;  
25         },  
26     });  
27  
28     $('.the-text').css({  
29         fontSize: function( index, value ) {  
30             return parseFloat( value ) * 1.15;  
31         },  
32     });  
33  
34     $('.highlight').css({  
35         fontSize: function( index, value ) {  
36             return parseFloat( value ) * 1.15;  
37         },  
38     });  
39 });
```

Fonte: Elaborada pelo autor (captura de tela).

**FIGURA 56** – Código jQuery para botão diminuição de fonte

```
41 $( "#down-font-size" ).on( "click", function() {  
42     $('h1').css({  
43         fontSize: function( index, value ) {  
44             return parseFloat( value ) / 1.15;  
45         },  
46     });  
47  
48     $('.the-text').css({  
49         fontSize: function( index, value ) {  
50             return parseFloat( value ) / 1.15;  
51         },  
52     });  
53  
54     $('.highlight').css({  
55         fontSize: function( index, value ) {  
56             return parseFloat( value ) / 1.15;  
57         },  
58     });  
59 });  
60 });
```

Fonte: Elaborada pelo autor (captura de tela).

#### 9.1.4. JANELA DE LIBRAS

JQuery não possui uma biblioteca para janela de intérprete virtual de LIBRAS, porém é possível por meio da ferramenta fazer a instalação do HandTalk, o mesmo aplicativo de LIBRAS mencionado na seção 7.1.1.2. Hand Talk - WordPress. Após entrar no site oficial e fazer o cadastro como indica o passo a passo, você receberá um token. Ele é uma chave que

será inserida no código em jQuery. Insira o código, conforme mostrado na imagem abaixo, na tag <script>.

**FIGURA 57** – Caixa de propriedades do token de *Hand Talk*

```
<script src="//api.handtalk.me/plugin/latest/handtalk.min.js"></script>
<script>
  var ht = new HT({
    token: "SEU TOKEN FICA AQUI"
  });
</script>
```

Fonte: Elaborada pelo autor (captura de tela).

### 9.1.5. CONCLUSÕES DA BIBLIOTECA

Com o jQuery, foi encontrada uma biblioteca i18n para contemplar a *feature* de multilinguismo, porém ela não é autossuficiente. Ela precisa de algumas outras ferramentas e bibliotecas para sua execução. Usar outra ferramenta que não seja jQuery não faz parte do foco de estudo deste projeto. Logo, concluímos que não é possível desenvolver uma ferramenta de multilinguismo apenas com jQuery.

Assim como multilinguismo, a ferramenta de pular para o conteúdo não foi possível ser desenvolvida com jQuery por falta de informação no repositório no gitHub. Veja no quadro abaixo a possibilidade de acessibilidade de uma biblioteca como essa:

**QUADRO 3** – Resultado da capacidade de acessibilidade do jQuery

	jQuery
Aumento e diminuição de fonte	SIM
Contraste de cores	SIM
Pular para o conteúdo principal	Não
Destacar e sublinhar <i>links</i>	Sim
Janela com intérprete virtual de LIBRAS	Sim
Multilinguismo	Não

Fonte: Elaborado pelo autor.

## 10. CONSIDERAÇÕES FINAIS

Concluimos que há várias formas de tornar um *site* acessível, seja um programador com conhecimento básico, intermediário ou avançado em codificação. Cada nível foi contemplado com uma ferramenta que visava o nível de complexidade no manuseio do usuário. Um Sistema de Gerenciamento de Conteúdo para usuários de nível básico, um *framework* para um usuário de nível intermediário e uma biblioteca para um usuário de nível avançado. Cada forma de acessibilidade foi trabalhada com uma ferramenta específica com sua devida explicação de uso na metodologia deste trabalho.

Para o CMS foi escolhido o WordPress, para o *framework* foi escolhido o Bootstrap e para a biblioteca foi escolhido o jQuery. Cada uma dessas ferramentas possui suas vantagens e desvantagens bem como suas limitações. Depois de analisá-las, veremos abaixo um quadro que reúne o resultado encontrado em cada respectivo capítulo das ferramentas:

**QUADRO 4** – Resultado final da capacidade de acessibilidade do WordPress, Bootstrap e jQuery

	WordPress	Bootstrap	jQuery
Aumento e diminuição de fonte	Sim	Não	Sim
Contraste de cor de fundo	Sim	Não	Sim
Pular para o conteúdo principal	Não	Sim	Não
Destacar e sublinhar <i>links</i>	Sim	Não	Sim
Janela com intérprete virtual de LIBRAS	Sim	Não	Sim
Multilinguismo	Sim	Sim	Não

Fonte: Elaborado pelo autor.

Com a clareza e simplicidade do CMS, vimos que foi possível contemplar quase toda a lista criada para este trabalho de *features* de acessibilidade. Isso significa que até mesmo uma pessoa sem nenhum conhecimento em programação é capaz de criar um *site* fortemente acessível para pessoas cegas, surdas e estrangeiras. Lembrando que é facilmente possível suprir a *feature* de pular para o conteúdo principal com um *site* estilo *one-page*.

O Bootstrap por sua vez nos mostra que infelizmente não é autossuficiente para deixar um *site* fortemente acessível. Contudo, devemos dar o devido reconhecimento de sua possibilidade de desenvolver a *feature* de multilinguismo. Vimos neste projeto que acessibilizar o conteúdo para vários idiomas resulta em um gigantesco aumento de possíveis acessos de pessoas do mundo inteiro.



Com a biblioteca jQuery, ao contrário do *framework*, foi possível contemplar quase todas as *features* da lista. Sua flexibilidade em alterar o CSS do documento *web* durante a execução do *site* foi primordial para sua competência e, embora a sua falta de autossuficiência para o desenvolvimento da *feature* de multilinguismo, devemos nos lembrar de que é possível criá-la com o auxílio de outras ferramentas, como outras bibliotecas também desenvolvidas sob a linguagem de JavaScript.

É possível compreender pelo quadro que as ferramentas que exigem um certo conhecimento em programação não contemplem por completo a lista de *features* criada para este trabalho. Porém, a programação é uma área fascinante que exige do seu profissional constante aprendizado para solução de problemas como este, encontrado pela análise do quadro, por isso é muito normal e até aconselhável que um profissional bem capacitado conheça mais de uma linguagem e ferramenta de desenvolvimento *web*. Se combinarmos o *framework* Bootstrap com a biblioteca jQuery seremos capazes de criarmos a lista de *features* de acessibilidade por completo e, por fim, teremos um *site* fortemente acessível para pessoas cegas, surdas e estrangeiras feito em código.

Conforme vimos nos capítulos anteriores, é extremamente importante que desenvolvamos tanto ferramentas que diminuam a distância entre informação e usuário bem como uma organizada arquitetura dos componentes de um *site*. A acessibilidade na *web* não pode ser reduzida a implantação de algumas ferramentas sem nenhum planejamento e instrução. Vimos que o gerenciamento de conteúdo de uma página na *internet* deve fazer parte da elaboração inicial de um *site*. Uma arquitetura previamente estruturada facilitará tanto o trabalho de um profissional quanto, principalmente, a navegabilidade de um usuário.

Por fim, ainda existem muitos sites que promovem pouco ou nenhum tipo de acessibilidade e assim podemos perceber a necessidade de trabalhos como este para promover melhores oportunidades para todos e para que a falta de informação, falta de tempo ou o alto custo não sejam mais usados como desculpas pelos desenvolvedores *web*. Por meio deste projeto foi possível entender também que ainda há outras formas de ferramentas para acessibilidade. Seria impossível inserir todas elas em um único trabalho de conclusão de curso, mas é interessante saber que elas também estão acessíveis na *internet* para que outros programadores possam se aventurar em usá-las. As ferramentas de acessibilidade existem e estão disponíveis para serem usadas.

## REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. Norma Brasileira, ABNT 9050. **Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos**, 2. ed. 2004. Disponível em: <<http://pfdc.pgr.mpf.mp.br/atuacao-e-conteudos-de-apoio/legislacao/pessoa-deficiencia/norma-abnt-NBR-9050>>. Acesso em: 19 nov. 2017.

BOIKO, Bob. **Content Management Bible**. Nova York, NY, EUA: John Wiley & Sons, Inc., 2001.

BRASIL. **Constituição Brasileira**. Decreto Federal Nº 5.296 de 2 de Dezembro de 2004, artigo 8. Dispõem sobre: “Regulamenta as Leis nºs 10.048, de 8 de novembro de 2000, que dá prioridade de atendimento às pessoas que especifica, e 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida, e dá outras providências.” Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/\\_ato2004-2006/2004/decreto/d5296.htm](http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2004/decreto/d5296.htm)>. Acesso em: 27 nov. 2017.

BRASIL. **Constituição Brasileira, Estatuto da Pessoa com Deficiência**. LBI – Lei 13.146/15. Art. 63. Dispõem sobre: “Institui a Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência).” Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2015/lei/l13146.htm](http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm)>. Acesso em: 27 nov. 2017.

BUCKLER, Craig. **Best JavaScript Frameworks, Libraries and Tools to use in 2017**, 2017. Disponível em: <<https://www.sitepoint.com/top-javascript-frameworks-libraries-tools-use/>>. Acesso em: 21 out. 2017.

CHAGAS, Fernando. **Um estudo sobre os sistemas de gerenciamento de conteúdo de código aberto**, Universidade Federal de Goiás. 2008. Disponível em: <[http://www.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF\\_002-08.pdf](http://www.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_002-08.pdf)>. Acesso em: 20 nov. 2017.

CLINTON, Marc. What is a Framework? **Code Project**, 2003. Disponível em: <<https://www.codeproject.com/Articles/5381/What-Is-A-Framework>>. Acesso em: 07 out. 2017.

CONFORTO, Débora; SANTAROSA, Lucila M. C. Acessibilidade à Web: Internet para Todos. **Revista de Informática na Educação: Teoria, Prática – PGIE/UFRGS**. v.5 nº 2. nov/2002, p.87-102.

FERNANDES, Jorge; GODINHO, Francisco. **Acessibilidade aos sítios Web da AP para cidadãos com necessidades especiais**, 2003. Disponível em: <<http://www.acessibilidade.gov.pt/manuais/manualv2.doc>>. Acesso em: 19 nov. 2017.

FERREIRA, Carlos A. R. **Internet**. Universidade Técnica de Lisboa, 2002. Disponível em: <<http://home.fmh.utl.pt/~cferreira/pdf/Internet.pdf>>. Acesso em: 18 nov. 2017.

GITHUB. **Paypal Accessibility**, 2015. Disponível em: <<https://github.com/paypal/bootstrap-accessibility-plugin>>. Acesso em: 01 set. 2017.

GOUVEIA, Luis M. B. **Sociedade da Informação**. Universidade Fernando Pessoa, 2004.

HAND TALK. **Oferecemos tradução digital para acessibilidade em LIBRAS, a Língua Brasileira de Sinais**, 2017. Disponível em: <<https://handtalk.me/>>. Acesso em: 22 abr. 2017.

INTERNET WORLD STATS. **Top Ten Languages Used in the Web**. Disponível em: <<https://www.internetworldstats.com/stats7.htm>>. Acesso em: 19 nov. 2017.

\_\_\_\_\_. **World Internet Users and 2017 Popular Statistics**. Disponível em: <<http://www.internetworldstats.com/stats.htm>>. Acesso em: 18 nov. 2017.

LEAL, Luciana; THOMÉ, Clarrisa. Brasil tem 45,6 milhões de deficientes. Jornal Estadão. Agencia Estado. Disponível em: <<http://www.estadao.com.br/noticias/geral,brasil-tem-45-6-milhoes-de-deficientes,893424>>. Acesso em: 19 nov. 2017.

MENEZES, Cláudio S. O multilinguismo e as novas tecnologias das Línguas no século XXI. **Belas Infieis**, v. 4, n. 1, p. 85-98, 2015.

MORVILLE, Peter; ROSENFELD, Louis. **Information Architecture for the World Wide Web**. 3. Ed. Editora O'Reilly Media, 2007.

NUNES, Sérgio S. **A Acessibilidade na Internet no Contexto da Sociedade da Informação**. Universidade do Porto, 2002. Disponível em: <<https://web.fe.up.pt/~mgi01016/is/acessibilidade.pdf>>. Acesso em: 19 nov. 2017.

NUNES, Sérgio S. **Introdução aos computadores**. Universidade do Porto, s.d. Disponível em: <<https://web.fe.up.pt/~ssn/disciplinas/crc/computadores.pdf>>. Acesso em: 18 nov. 2017.

NV ACCESS. **Empowering lives through non-visual access to technology**, 2017. Disponível em: <<https://www.nvaccess.org/>>. Acesso em: 18 nov. 2017.

O'REILLY. **The History of Programming Languages**, 2017. Disponível em: <[http://archive.oreilly.com/pub/a/oreilly/news/languageposter\\_0504.html](http://archive.oreilly.com/pub/a/oreilly/news/languageposter_0504.html)>. Acesso em: 25 out. 2017

PASSOS, Rafaela L. **Um aplicativo multilíngue voltado a estrangeiros em intercâmbio na Universidade de Brasília**. Universidade de Brasília, Brasília/DF, 2017.

SANTOS, Youseph. **Escrevendo HTML semântico**, 2016. Disponível em: <<http://maujob.com/tutorial/escrevendo-html-semantico.php>>. Acesso em: 12 nov. 2017.

SOUZA, Moisés de. Redes de Computadores. **Acervo Digital**. Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância, Unidade Acadêmica de Educação a Distância e Tecnologia - Universidade Federal Rural de Pernambuco. Recife: Acervo Digital EADTEC, 2011. Disponível em: <<http://www.ead.ufrpe.br/acervo-digital-eadtec/node/665>>. Acesso em: 20 nov. 2017.

TECMUNDO. **Marco histórico: 25% dos sites na internet são feitos em WordPress**. Disponível em: <<https://www.tecmundo.com.br/wordpress/89114-marco-historico-25-sites-internet-feitos-wordpress.htm>>. Acesso em: 20 nov. 2017.

WATRALL, Ethan; SIARTO, Jeff. **Head First Web Design**. Editora O'Reilly Media, 2009.

WEB ACCESSIBILITY INITIATIVE. **Introduction to Web Accessibility**, 2005. Disponível em: <<https://www.w3.org/WAI/intro/accessibility.php>>. Acesso em: 19 nov. 2017.

WIKIPEDIA. **Bootstrap** (front-end framework), 2017. Disponível em: <[https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))>. Acesso em: 21 nov. 2017.

\_\_\_\_\_. **Bibliotecas** (computação), 2016. Disponível em: <[https://pt.wikipedia.org/wiki/Biblioteca\\_\(computa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Biblioteca_(computa%C3%A7%C3%A3o))>. Acesso em: 21 out. 2017.

\_\_\_\_\_. **Rede de Fornecimento de Conteúdo**, 2017. Disponível em: <[https://pt.wikipedia.org/wiki/Rede\\_de\\_fornecimento\\_de\\_conte%C3%BAdo](https://pt.wikipedia.org/wiki/Rede_de_fornecimento_de_conte%C3%BAdo)>. Acesso: 21 nov. 2017.

\_\_\_\_\_. **GitHub**, 2017. Disponível em: <<https://pt.wikipedia.org/wiki/GitHub>>. Acesso em: 20 nov. 2017.

\_\_\_\_\_. **Modelo de Objeto de Documentos**, 2017. Disponível em: <[https://pt.wikipedia.org/wiki/Modelo\\_de\\_Objeto\\_de\\_Documentos](https://pt.wikipedia.org/wiki/Modelo_de_Objeto_de_Documentos)>. Acesso: 21 nov. 2017.

WORDPRESS. **AKISMET**, 2017. Disponível em: <<https://wordpress.org/plugins/akismet/>>. Acesso em: 19 nov. 2017.

\_\_\_\_\_. **Contact Form 7**, 2017. Disponível em: <<https://wordpress.org/plugins/contact-form-7/>>. Acesso em: 19 nov. 2017.

\_\_\_\_\_. **FLARE**, 2017. Disponível em: <<https://wordpress.org/plugins/flare/>>. Acesso em: 19 nov. 2017.

\_\_\_\_\_. **Google Language Translator**, 2017. Disponível em: <<https://wordpress.org/plugins/google-language-translator/>>. Acesso em: 22 ago. 2017.

\_\_\_\_\_. **Hand Talk - WordPress**, 2017. Disponível em: <<https://wordpress.org/plugins/handtalk/>>. Acesso em: 22 ago. 2017.

\_\_\_\_\_. **Meet WordPress**, 2017. Disponível em: <<https://wordpress.org/>>. Acesso em: 20 nov. 2017

\_\_\_\_\_. **Plug-ins**, 2017. Disponível em: <<https://wordpress.org/plugins/>>. Acesso em: 19 nov. 2017

\_\_\_\_\_. **Smart Archives Reloaded**, 2017. Disponível em: <<https://wordpress.org/plugins/smart-archives-reloaded/>>. Acesso em: 19 nov. 2017.

\_\_\_\_\_. **WordPress.com is the easiest place to get started**, 2017. Disponível em: <<https://wordpress.com/>>. Acesso em: 20 nov. 2017.

\_\_\_\_\_. **WP Accessibility Helper**, 2017. Disponível em: <<https://wordpress.org/plugins/wp-accessibility-helper/>>. Acesso em: 22 ago. 2017.

\_\_\_\_\_. **W3 Total Cache**, 2017. Disponível em: <<https://wordpress.org/plugins/w3-total-cache/>>. Acesso em: 19 nov. 2017.

WORLD WIDE WEB CONSORTIUM. **W3C**. Disponível em: <<https://www.w3.org/>>. Acesso em: 25 set. 2017.